SP 907B4A

CALVERT COUNTY SNOW EMERGENCY DECISION SUPPORT SYSTEM

Final Report

Submitted to:

The State Highway Administration
Maryland Department of Transportation

Ali Haghani
Associate Professor of Civil Engineering &
Director, Transportation Program
Department of Civil Engineering
University of Maryland
College Park, MD 20742
Phone: (301) 405-1963
Fax: (301) 405-2585
Email: haghani@eng.umd.edu

August 2000

Summary and Acknowledgements

This report summarizes the results of research conducted to develop a decision support system for assisting the Maryland State Highway Administration Office of Maintenance staff in designing snow emergency routes for Calvert County, MD. The research deals with the problem of designing efficient routes for salting and plowing trucks in snow emergencies. The problem is formulated as a mathematical optimization problem and is classified as a Capacitated Rural Postman Problem that is an arc routing problem. The capacitated arc routing problem has been shown to be NP-hard. Several heuristic algorithms are proposed and a combination of these procedures is used to solve the real-world snow emergency vehicle routing problems in Calvert County. The results of the implementations indicate that there is room for improvements in service and savings in operational costs.

The research team wishes to express its sincere appreciation and gratitude toward the Maryland State Highway Administration Office of Maintenance personnel whose help in gathering the necessary data for the project, and whose comments and suggestions was an invaluable asset in the course of the project. Special thanks go to Mr. Charles L. Hall, Resident Maintenance Engineer, Mr. Russell A. Yurek, Deputy Chief Engineer, and Mr. Fran McGrath, the Project Manager.

TABLE OF CONTENTS

1. Background

1.1 Route Design in Snow Emergency Salting and Plowing Operations

In Unite States, large amounts of money are invested in snow removal and disposal operation. According to SHRP, snow and ice control operations cost State and local highway agencies in excess of $1.8 billion annually (1994). Eliminating or reducing snow and ice on the pavements and bridges improve the safety of our highways. However, many planning and routing decisions are not so efficient. Efficient route design and planning for the snow vehicles can not only save highway agencies' operational cost, but also improve level of service and social benefits.

Snow removal and disposal problem is a multi-objective problem. It includes many sub-problems such as which and how much deicer to use, where to locate the depots, how to assign the routes to the vehicles, how to schedule the personnel, etc.

This research focuses on how to assign the routes to the vehicles. A good route assignment can improve the level of efficiency. The efficiency is defined in terms of the amount of time spent treating the roads as a percentage of the total amount of time spent on the route. Figure 1.1 shows the relationship between the time and cost of the operation (Waddell, 1994). Reaction time is the time interval between the weather first beginning to adversely impact the roads and the decision to initiate the snow and ice control operations. Preparation time is the time interval between the decision to start the operations and the beginning of actual road clearing. Execution time is the time interval from the beginning of clearing to the end of the operation. A good routing plan that can reduce the operation time will reduce the operational costs and social costs, shown as Figure 1.2.

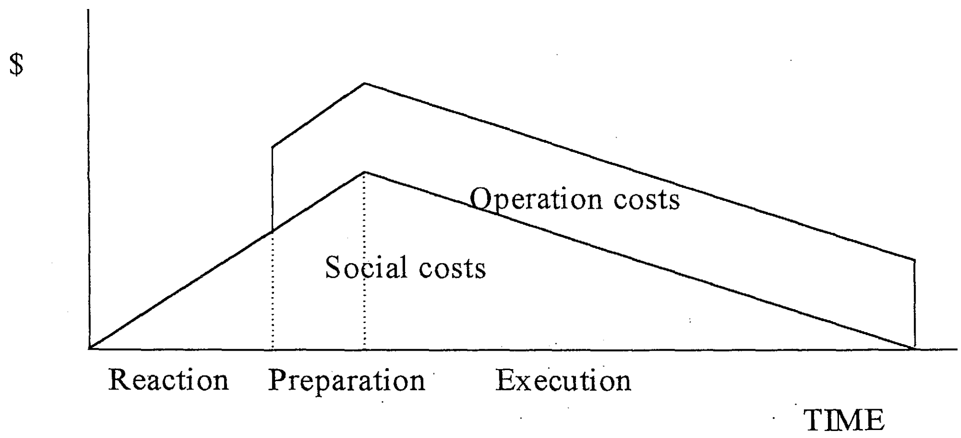


Figure 1.1 Time vs. Cost

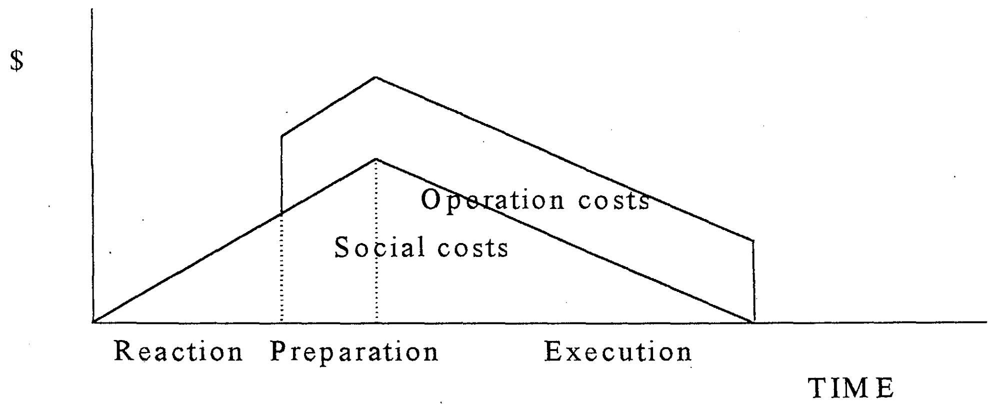


Figure 1.2 Time vs. Cost (Good Route Planning)

1.2 The Nature of the Problem

The objectives of the snow route planning are to find the best route for each vehicle, minimize the total deadhead time and minimize the number of trucks. Snow vehicle routing problems have many features.

First, snow route design can be classified into the category of Arc Routing Problems (ARP) in which the vehicles should service some or all arcs in a network. More precisely, the problem is a Rural Arc Routing Problem in which only a subset of the network arcs needs to be serviced.

Second, in snow emergency operations we are dealing with a directed network in which every link in the network has a direction. Although most roadways are two-way roads, there are still some one-way roads. In this situations the trucks must service both directions for most roadways. Also, the underlying network is a hierarchical network. This means that there is a precedence structure for servicing the roads. Some roads that are more important should be serviced first. The less important ones can be serviced later.

Golden and Wong (1981) formulated the Capacitated Arc Routing Problem (CARP) that is very close to the snow operation route design problem and suggested many algorithms for that problem. However, a major deficiency of this formulation is that it only deals with non-directed networks. This simplification makes the problem much easier to deal with. Neither Golden's nor any other existing algorithms in the literature can be applied to the capacitated arc routing problem on directed networks, which is a much more difficult problem. Although some researchers suggest approaches to deal with this problem, no mathematical formulation for this problem has been proposed in the literature.

As the name suggests, there are vehicle capacity constraints in this problem. For salting, the trucks can only load a certain amount of salt. When they run out, they have to go back to the depot to load more salt. An important factor that must be considered is that the vehicle capacities are generally different. Many existing approaches assume that the vehicle capacities are the same. In reality, state agencies operate many different types of trucks with different capacities. Therefore for a realistic formulation, we cannot assume that the vehicle capacities are the same. This makes the problem much more difficult.

There may also be some other operational constraints that a truck cannot travel more than a certain amount of time or distance because the driver needs to rest and the truck must refuel.

In general, the route design in snow emergency operations can be described as follows. Given a directed transportation network G (V, A) with arc demands q_{ij} for each arc (i, j) $\in$ A, that must be satisfied by a fleet of vehicles k, each of which have a capacity W_k, and starting service time constraints (if any), find the number of cycles which all pass through the domicile and satisfy the demands at minimal total cost.

Several related problems are as follows:

(1) The Chinese Postman Problem (CPP) that finds a minimum-cost cycle that traverses every arc in the network at least once;

(2) The Rural Postman Problem (RPP) that finds a minimum-cost cycle that traverses each arc in a given subset of the arcs in the network at least once;

(3) The Capcitated Chinese Postman Problem (CCPP) in which, given arc demands $q_{ij} > 0$ for each arc (i, j) that must be satisfied by vehicles of capacity W_k, the objective is to

find a set of cycles that all pass through the domicile and satisfy demands at minimal total cost;

(4) The Traveling Salesman Problem (TSP) that finds a minimum-cost tour that visits each mode in N exactly once;

(5) The Vehicle Routing Problem (VRP) that finds the minimum cost routes that service all nodes with demand in the network exactly once; and,

(6) The General Routing Problem (GRP) that finds the minimum cost routes that service a sub set of nodes and arcs in the network.

1.3 Research Objectives and Scope

This research focuses on the salting operation. While we will look at the plowing problem as well, the intent is not to solve both problems. Because of the nature of the snow emergency vehicle routing problem, Golden's formulation cannot be applied here. In this research the salting problem is formulated as a Capacitated Rural Arc Routing Problem in a directed network. This problem will be termed the Capacitated Rural Directed Arc Routing Problem (CRDARP). The proposed formulation will also deal with some other features of the salting problem including the travel time constraints and service level constraints.

As mentioned before, because of the different nature of the problem, the existing solution methods in the literature cannot be applied to this problem directly. This research proposes an algorithm to solve this problem. This algorithm is applied to solve the salting problem. Although the proposed algorithm is only used to solve the salting problem, with minor modifications, it can also be applied to plowing problems. The proposed algorithm can also be viewed as a contribution to solving the general arc routing problems.

1.4 Organization of the Report

The rest of this Report is organized as follows. Section 2 provides a comprehensive review of the literature in all related problems. Section 3 presents the mathematical formulation of the problem and discusses the heuristic methods proposed for solving the salting problem and their advantages and disadvantages. In Section 4 the results of the algorithm implementation in Calvert County, Maryland are presented. Finally, Section 5 presents the conclusions.

2. Literature Review

Dealing with snow emergencies involves a number of different operations. Campbell and Langevin (1994) provide a detailed review of the snow removal and disposal operation. Cook and Alprin (1976) proposed a salt spreader truck routing heuristic algorithm based on closest street selection to minimize the time to cover all branches in a network. Eglese (1994) presented a heuristic to minimize the distance traveled by gritting vehicles. Haslam and Wright (1991) discussed the strengths and weaknesses of several mathematical programming approaches and present a multiple objective heuristic methodology for the design of routes for intrastate highway snow and ice control.

Liebling (1973) presented a study done for the city of Zurich, based on a CPP procedure and a heuristic to partition the city between the vehicles. Marks and Stricker (1971) suggested a cluster-first and route second heuristic that uses a CPP model for route. Gilbert (1990) modeled snow blower routing as an asymmetric CPP with duration, precedence and time window constraints and developed an insertion heuristic. Gelinas (1992) described an optimal dynamic programming solution procedure for the CPP with precedence relation and included an application to snow plowing in Montreal. Simulation studies (Tucker & Clohan, 1973; England, 1982) have also been conducted to evaluate routes produced by different procedures.

From these studies we can conclude that snow emergency vehicle routing problem is most widely considered as a network optimization problem. In fact, most snow removal and disposal problems are dealt with as arc routing problems. In this research, we will modify and suggest several constructive and improvement heuristic method for Act Routing Problem. Various types of network optimization problems are highly related and algorithms developed for one problem, with minor modifications, can often be used to solve another problem. Therefore, it is very helpful to review the existing literature in all related problems before we get into our specific problem.

2.1 Node Routing Problems

Node routing problems try to find the minimum cost routes that service the nodes in the network. There are two main problems in this category, one is the vehicle routing problem and the other is the traveling salesman problem.

2.1.1 Vehicle Routing Problem (VRP)

The classical vehicle routing problem (VRP) can be described as follows: A fleet of m capacitated vehicles localized at one or more depots have to serve n customers with demand d_i. Let G = (N, A) be a graph where N = $\{(i, j)\ i, j \subseteq N, i \neq j\}$ is the arc set. Vertex 0 represents a depot at which a fleet of m vehicles is based. The remaining vertices correspond to the customers. Let each vehicle v have a capacity equal to q_v, every vertex i of N have a non-negative demand $q(i) \leq \text{Max}\ \{q_v\}$, and every arc { i, j } have an associated non-negative distance or cost c(i, j). The problem is to find the minimum cost route for all vehicles so that the customer demands are satisfied.

2.1.2 Traveling Salesman Problem (TSP)

In classical vehicle routing problem if we have only one vehicle and there is no capacity constraint, the problem simplifies into a traveling salesmen problem. In fact, many methods devised to solve VRP's involve solving TSP's as sub-problems. traveling salesman problem can be described as follows. Consider the situation of sequencing n jobs on a single facility. A setup cost is incurred every time a new job is scheduled, but this setup cost depends on the immediately preceding job that was processed on the facility. The objective is to determine the sequence that minimizes the total sum of setup costs.

Both VRP and TSP are typical NP-complete problems. These problems have attracted many researchers over the past decades. They are highly related to each other and an algorithm developed to solve one, can always be applied to the other problem.

2.2 Arc Routing Problem

The arc routing problems belong to another subset of the network optimization problems. While in node routing problems we are trying to visit the nodes of the network, in arc routing problems the objective is to traverse the arcs of the network.

2.2.1 Chinese Postman Problem (CPP)

Chinese Postman Problem is simply stated by Guan (1962) as the problem of finding the shortest walking distance for a mailman who has to cover his assigned segment before returning to the post office. There are two extensions of the CPP, one is windy postman problem (WPP), in which the underlying network is an undirected graph, but the cost of traversing an arc depends on the direction of travel. Another form of the CPP is the hierarchical CPP where a precedence relation is defined on arcs of the graph, and the order in which the arcs are serviced must respect this relation. The CPP can be viewed as the counterpart of the TSP in the category of arc routing problems.

2.2.2 Capacitated Chinese Postman Problem (CCPP)

As a counterpart of VRP in the arc routing problems, the capacitated Chinese postman problem deals with a more realistic case than CPP. Given arc demands q (i, j)>0 for each arc (i, j) which must be satisfied by vehicles of capacity W_i, find a set of cycles which all pass through a domicile and satisfy demands at minimal total cost.

2.2.3 Rural Postman Problem (RPP)

Just like Chinese postman problem that is more likely to arise in urban areas, the RPP is commonly associated with mail delivery in rural areas. There are a number of areas whose set of streets has to be serviced by a postman, and the other set of links between those areas that do not have to be served, but may be used for traveling between those areas. The problem is to find the minimum cost route to service those arcs that need to be served.

2.3 General Routing Problem (GRP)

General routing problem is a relaxation of the vehicle routing problem and the arc routing problem. Given a network G (N, E, C), where N is the set of the nodes, E is the set of arcs, and C is the matrix of costs; the problem is to find the minimum-cost cycle that visits every node in subset $Q \in N$ and traverses every arc in subset $R \subseteq E$. From this definition, we can see that both node routing problems and arc routing problems are special cases of the GRP.

(1) If $Q = \Phi$ and $R \subset E$, then the GRP is the rural postman problem.

(2) If $Q = \Phi$ and $R = E$, then the GRP is the Chinese postman problem.

(3) If $Q = N$ and $R = \Phi$, then the GRP is the vehicle routing problem or traveling salesman problem depending on the capacity constraints.

From the above analysis we can see that the node routing problems and the arc routing problems are very closely related. Because the node routing problems have received more attention than the arc routing problems, many scholars have tried to use the algorithms that were developed for the node routing problems for solving the arc routing problems and there have been some successes.

2.4 Review of the Solution Procedures for the Vehicle Routing Problems

Vehicle routing problems are typical routing problems. Several exact and heuristic algorithms have been proposed to solve different kinds of vehicle routing problems. We will review some of these procedures.

2.4.1 Exact Algorithms

There are four exact algorithms for solving the VRP. We can treat the traditional VRP, which has time window constraints, as a special case of the general VRP that has no time window constraints.

Approaches Based on Dynamic Programming

These approaches can be regarded as extensions of the Christofides (1981) state space relaxation method to problems with time windows. Problems with up to 15 customers have been solved to optimality.

Approaches Based on Column Generation

These algorithms are based on a combination of linear programming relaxed set covering and column generation. These are currently the best approaches. Desrochers et al. (1992) presented an exact method with capability of solving some 100-customer problems.

Lagrangian Decomposition Based Methods

These algorithms apply various Lagrangian decomposition schemes to VRP with time windows in order to produce lower bounds. They are currently capable of solving some 100-customer problems to optimality using a combination of Lagrangian decomposition.

K-Tree Based Methods

Fisher (1994) has extended the 1-tree method to K-tree method for the classical vehicle routing problem and VRP with time windows.

The first three approaches rely on the solution of a shortest path problem with time windows, and the vehicle capacity constraints are used either to generate columns or as a part of a Lagrangian relaxation.

The K-tree approach is an extension of the classical tree approach for the traveling salesman problem to the case with several capacity constrained vehicles. In this approach it is assumed that each route contains at least two customers. A special mathematical model is formulated and the constraints are relaxed. The approach can be extended to the case with time window constraints. Lagragian relaxation is still applied. The relaxed problem is a degree contained K-tree problem. The multipliers are determined using the subgradient method. New capacity and time window constraints are generated as they are violated.

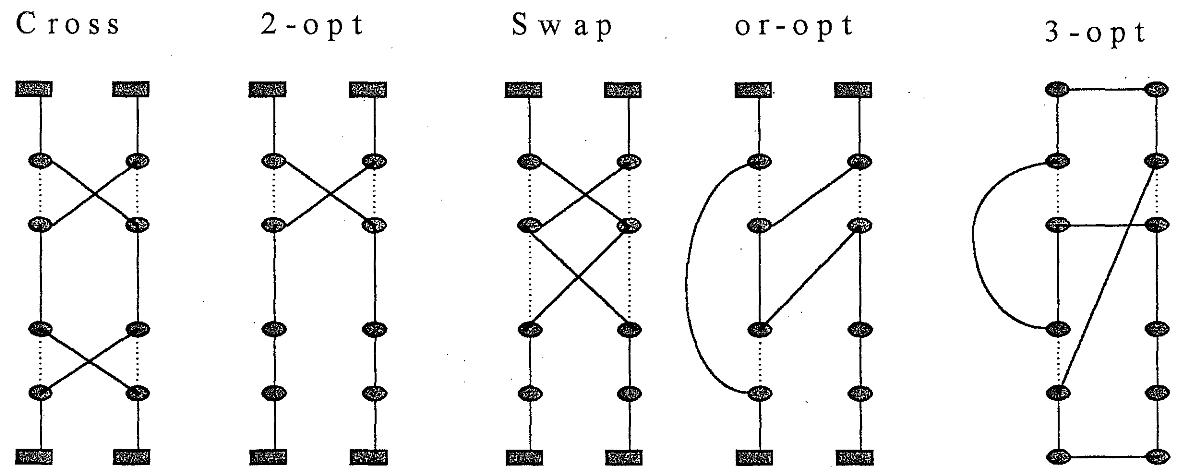
2.4.2 Route Construction Heuristics

These procedures build a solution following a number of steps. At each step a current configuration that is possibly feasible is compared with an alternative configuration that yields the largest saving in terms of some criterion function, or that inserts, least expensively, a demand entity not in the current configuration, into the existing routes.

2.4.3 Route Improvement Heuristics

These methods group or cluster demand nodes and/or arcs and design economical routes over each cluster. First a large route or cycle is constructed which includes all demand entities. Next, the large route is partitioned in to a number of smaller but feasible routes.

There are several kinds of route improvement heuristics. Each of them has some advantages and disadvantages. The most commonly used methods are shown in the following diagrams:



There is yet another type of heuristic algorithms. These are called composite heuristics. These heuristics mix both route construction and route improvement procedures. After constructing the feasible routes, one feasible solution is altered to yield another feasible solution with a reduced overall cost. These procedures continue until no additional cost reductions are possible.

2.5 Review of the Solution Procedures for the Arc Routing Problem

Because most arc routing problems are NP-hard, this review will focus more on the heuristic methods for the capacitated problems.

In his review article, Eiselt (1992) classified the existing heuristics for the capacitated arc routing problem into three categories, 1) simple constructive methods; 2) two-phase constructive methods; and 3) improvement methods.

2.5.1 Simple Constructive Methods

There are five constructive methods that are as follows.

<u>Constructive-Strike Algorithm</u>

This heuristic was first proposed by Christofides (1973) and improved by Pearn (1989). The basic concept of the algorithm is that it gradually constructs feasible cycles and removes them from the network. When a cycle is constructed, the algorithm tries to avoid the disconnected sub-graphs. When feasible cycles can no longer be found, an Euler cycle is constructed on the remaining graph. Pearn modified the algorithm in 1989, he suggested using the minimal spanning tree algorithm to render the remaining graph connected and the matching algorithm to generate an Euler cycle. The construct-strike, minimal spanning tree and matching procedures are repeated until the whole graph is covered.

<u>Path-Scanning Algorithm</u>

The basic idea of this algorithm is to construct one cycle at a time based on a certain optimization criterion. In forming each cycle, a path is extended by joining the arc that looks most promising until the vehicle capacity is exhausted. Then the shortest return path to the depot is followed. Golden, DeArmon and Baker (1983) suggested five criteria:

(a) the cost per unit remaining demand is minimized

(b) the cost per unit remaining demand is maximized

(c) the cost form node j back to the depot is minimized

(d) the cost from node h back to depot is maximized

(e) if the vehicle is less than half full, maximize the cost from node j back to the depot, otherwise minimize it

Each of these criteria is used to generate a complete solution, and the solution from this approach is the best of these five.

Pearn modified the path-scanning algorithm in 1989. Instead of using each single criterion to generate a complete solution, one of the five criteria is selected randomly to use in each step while extending the cycle path, and generating a complete solution. The best of the 30 solutions is then chosen as the best solution for this approach.

Augment-Merge Algorithm

Golden, DeArmon and Baker (1983) suggested this algorithm which was inspired by the Clarke and Wright algorithm for the vehicle routing problem (1964). Initially, all arcs belong to different cycles. Then, cycles are gradually merged according to a savings criterion.

The basic procedure is outlined below:

Step 1. Initialize -- all demand arcs are serviced by a separate cycle

Step 2. Augment -- starting with the largest cycle available, see if a demand arc on a small cycle can be serviced on a larger cycle

Step 3. Merge -- subject to capacity constraints, evaluate the merging of any two cycles. Merge the two cycles that yield the largest positive savings.

Step 4. Iterate -- repeat step3 until finished

The number of augment-merge program executions is limited to 24, and the solution from this approach is the best of the 24 runs.

In the proposed algorithm in this research, this algorithm is used in the directed network. Therefore, some modifications to this algorithm are made.

Parallel Inserting

Chapleau et al. (1984) proposed this algorithm. It is similar to the path-scanning algorithm but constructs several routes in parallel.

Augment-Insert Algorithm

Pearn (1991) proposed this algorithm. The first phase of the algorithm is trying to construct the least cost cycle that covers arcs and augment the initial cycle until the vehicle capacity exhausted. The second phase is inserting the remaining arcs into the existing cycles until all capacities are exhausted, the least cost return paths from the two end points to the depot form a complete cycle. The algorithm is specifically for solving problems with sparse networks and large arc demands. The algorithm requires upper bound value for insertion cost. Therefore, setting the upper bound value is also an issue.

2.5.2 Two Phase Constructive Methods

Borrowing the ideas from the vehicle routing problem, the two-phase constructive algorithms for CARP can be classified into two categories:

Cluster-First, Route-Second Heuristic

In this algorithm, the arcs are first partitioned into clusters, each having a total weight not exceeding W. For this, a greedy criterion can be applied or a generalized assignment algorithm can be used. Then a vehicle route is determined for each cluster by a simple modification of a CPP algorithm

Route-First, Cluster-Second Heuristic

This heuristic first constructs a giant Euler tour over all edges with positive demands. The tour is then partitioned into feasible clusters, this can be done by a bin packing heuristic, and a vehicle tour is constructed for each cluster.

2.5.3 Improvement Methods

These methods also borrow the ideas from vehicle routing algorithms. While there are some suggestions on the algorithms, there are few implementations of these suggestions. Typically, the improvement methods are inspired by edge exchange heuristics for the traveling salesman problem. Win (1987) suggested the use of simulated annealing. Jin Yuan Wang and Jeff R. Wright (1994) suggest using tabu search to improve the solution. So far no computation results are reported in the literature.

In this paper the focus is on developing improvement algorithms. Several improvement heuristics for CRDARP are proposed. These algorithms can be used as a basis to generate more advanced algorithms.

2.6 Conclusions

In this Section, the problems relevant to the topic of this research and the solution procedures for these problems were reviewed. From the literature review, we can see that most algorithms for the arc routing problems use the ideas of the algorithms for vehicle routing problems. Although arc routing problems have wider field applications, they have attracted less attention than vehicle routing problems. While many heuristics have been suggested for vehicle routing problem, few heuristics (especially improvement heuristics) have been used for arc routing problem. In this research, several improvement algorithms for arc routing problems are suggested. The ideas of edge exchange heuristics for the VRP are also used for the arc routing problem, but because the difference between the ARP and the VRP, the heuristics for the VRP are modified for the ARP.

3. Problem Formulation and Solution

3.1 Problem Formulation

3.1.1 Integer Programming Formulation

There are two integer Programming formulations for the CARP. Golden and Wong (1981) proposed the first formulation, and the other one is proposed by Belenguer and Benavent (1991). The first one uses variables defined on directed arcs and the second one uses variables defined on undirected arcs. Since a transportation network is a directed network, the proposed formation is mostly based on the formulation proposed by Golden and Wong.

Although the proposed formulation can be used for both salting and plowing cases, in this research we will just focus on the salting problem. For the salting problem there are truck capacity constraints. For the plowing problem, some network transformation is necessary.

The important point in the proposed formulation is that it accounts for the hierarchical network. Time windows are used to show the constraints that deal with the network hierarchy. If an arc has a higher priority, it must have a more strict time window for service that must be ahead of those arcs that have lower priorities. The following notation is defined:

G	=	(V, A) is a connected directed graph.
V	=	$\{v_i, \ldots\ldots, v_j\}$ is the vertex set
A	=	$\{(v_i, v_j): v\ , v_j \in V$ and $i \neq j\}$ is the arc set
X_{ijk}	=	1 if and only if edge (v_i, v_j) is <u>traversed</u> from v_i to v_j on route k and is deadhead
	=	0 otherwise

Y_{ijk}	=	1 if and only if edge (v_i, v_j) is <u>serviced</u> from v_i to v_j on route k
	=	0 otherwise
s_{ijk}	=	Arrival time for arc (i, j) on route k
t_{ijk}	=	Service time for arc (i, j) on route k
t'_{ijk}	=	Travel time for arc (i, j) on route k
C_{ij}	=	Cost associated with arc (i, j)
W_k	=	Capacity of vehicle k
q_{ij}	=	Arc (i, j) demand
K	=	A large number
n_{ij}	=	The number of times that the arc (i, j) should be serviced
		Note: if arc (i, j) is a two lane highway, n_{ij} should equal to ½, if the arc can be serviced by traveling in one direction only
b_{ij}	=	Beginning service time for arc (i, j)

The problem formulation is as follows:

Minimize the deadhead distance

$$\text{Min} \quad \sum_{k=1}^{m} \sum_{(v_i,v_j)\in A} C_{ij} X_{ijk}$$

Subject to:

For each node, the in-degree should equal to out-degree

$$\sum_{(v_i,v_j)\in A} X_{jik} - \sum_{(v_i,v_j)\in A} X_{ijk} + \sum_{(v_i,v_j)\in A} Y_{jik} - \sum_{(v_i,v_j)} Y_{ijk} = 0 \qquad (v_i \in V,\ k = 1, \ldots, m)$$

For two-lane highways, if they can be salted in one direction, the sum of both directed arc variables should be 1.

$$\sum_{k=1}^{m}(Y_{ijk}+Y_{jik})=1 \qquad \text{if both } n_{ij} \text{ and } n_{ji} \text{ equal to } ½ \text{ for all } (v_i,v_j)\in A$$

Each arc should be serviced as required:

$$\sum_{k=1}^{m}(Y_{ijk})=n_{ij} \qquad \text{if } n_{ij} \text{ is greater than 1 for all } (v_i,v_j)\in A$$

Capacity constraint (only used in salting problem)

$$\sum_{(v_i,v_j)\in A} q_{ij}Y_{ijk} \le W_k \qquad (\text{k} = 1, \ldots, \text{m})$$

Vehicle k cannot arrive at j before s_{ij} plus the service time or travel time along the arc (i, j)

$$s_{ijk} + t_{ijk} * Y_{ijk} + t'_{ijk} * X_{ijk} - \text{K}(1- X_{ijk} - Y_{ijk}) \le s_{jlk} \quad \forall k \in (1,2,3,\ldots) \text{ all } (v_i,v_j)\in A$$

Time constraints for starting service time for each node

$$s_{ijk} - K(1-Y_{ijk}) \le b \qquad \text{for all } \forall k \in (1,2,3,\ldots)$$

Total travel time constraints

$$\sum_{(v_i,v_j)\in A}(t_{ijk} * Y_{ijk} + t'_{ijk} * X_{ijk}) < t_k \qquad \text{for all k.}$$

All vehicles should start from the depot

$$\sum_{j\in N}(X_{0jk}+Y_{0jk})=1 \quad \forall k \in \text{v}$$

All vehicles should end at the depot

$$\sum_{j\in N}(X_{in+1k}+Y_{in+1k})=1 \quad \forall k \in \text{v}$$

Sub-tour elimination constraints

$$\sum_{v_i, v_j \in S} (X_{ijk} + Y_{ijk}) \leq |S| - 1 + n^2 u_k^s \quad S \in (V \setminus depot)$$

$$\sum_{v_j \in S} \sum_{v_j \in S} (X_{ijk} + Y_{ijk}) \geq 1 - w_k^s$$

$$u_k^s + w_k^s \leq 1$$

$$u_k^s, w_k^s \in \{0,1\}$$

$$X_{ijk}, Y_{ijk} \in \{0,1\} \qquad ((v_i, v_j)) \in A; k = 1, \ldots, m)$$

The sub-tour elimination constraints ensure that the solution does not contain any illegal sub-tours. Golden and Wong (1981) has the detailed description of how sub-tours are broken.

Note that for the plowing problem, the capacity constraints are not necessary. But the network should be transformed to fit this formulation. There are several major differences between the proposed formulation and that of Golden and Wong (1981). These differences are described in detail in the following section.

3.1.2 The Improvements Compared to Golden and Wong Formulation

Golden and Wong (1981) tried to solve the more general problem and therefore, their formulation is a more general one. It just has the capacity constraint, and assumes that each arc just needs to be serviced only once. In an attempt to use this formulation for the real-world snow emergency routing problem, several special considerations should be taken into account in the formulation.

The first difference is the required number of times of service for the links in the network. Golden and Wong assume that each arc needs to be serviced just one time. In reality, if the road has more than 2 lanes in each direction, in salting operation, the truck should salt the road more than once. Another problem is that some two-lane highways can be serviced only once from one direction. That means when the truck traverses the highway in one direction and spreads the salt once it covers both lanes of the highway. Dealing with this situation requires formulating the problem on a mixed network that will make the problem significantly more difficult.

Another difference is the capacity constraint. In Golden and Wong formulation, the vehicle capacities are homogeneous. In reality, the vehicle capacities are not the same. To respond to snow emergencies, the State Highway Administration uses its own trucks as well as rented trucks. These trucks all have different capacities for hauling salt. There are two kinds of trucks usually used in salting, one has the capacity of 8 tons, the other one 15 tons. Special considerations should be given to the truck capacities and identification of the type of truck that is needed to serve a particular route. These considerations make the problem more difficult to solve that means the solution times can be longer.

The third difference is the time window constraints. Since more important routes need to be serviced first and less important ones can be serviced later, we are dealing with a hierarchical network. The formulation proposed in this research uses time window constraints to deal with the hierarchy of the network. We can set the time windows for the routes that need to be serviced first. These time windows can be stricter or narrower than the other routes. The time window constraints are very necessary for salting and plowing operation. At the present time, Maryland State Highway Administration does not have such standards. Therefore, in the case study presented in this research we do not incorporate these constraints. However, because of their importance in the more general snow emergency response vehicle routing problem they are included in the proposed formulation.

The first difference between the proposed formulation and Golden and Wong formulation does not exist when we are dealing with a non-directed network because there is no direction difference. Most algorithms that are suggested thus far deal with the non-directed network, and therefore, are not capable of handling the situation in directed networks as described above. The second difference makes the proposed formulation more practical. The time window constraints also add more practical sense, although currently there are no such standards for level of service in existence.

3.2 Solution Algorithm for the Salting Problem

3.2.1 The Model and Its Properties

As we discussed before, the nature of the salting problem makes it a much more difficult problem. Since the salting trucks can salt two lanes at one time, for two lane highways, the trucks need to just salt one time for both lanes. If a truck traverses the highway once and services it in one direction, it does not need to service the highway in the opposite direction. This makes the problem more difficult because we do not know which direction we should service. In an attempt to solve the problem we can try both directions. However, if there are many two-lane highways, there will be numerous combinations and the solution time expands exponentially. Another difficulty is the truck capacity. For plowing operations, there is no such constraint, but trucks must go back to depot to reload if they run out of salt in the salting operation. Also, the transportation network is a directed network. This also makes the applicability of other algorithms in this context very difficult. These existing algorithms cannot be easily transformed for application to the salting problem directly.

3.2.2 Solution Method Considerations

The salting problem is very similar to the capacitated arc routing problem, and the algorithms that are developed for those problems, as we reviewed before, can work in this context. Construct-strike algorithm works well for dense networks, and augment-insert algorithm is good for sparse networks. We propose a modification of Golden, DeArmon and Baker's algorithm (1983) to make it suitable for the directed network. We call this modified algorithm an exchange and merge algorithm. There are few studies that report that application of improvement methods results in improved routes. In the proposed algorithm we use an improvement method to enhance the results that are obtained by means of any route construction heuristics.

In the proposed algorithm, the network is initialized as follows. The network is composed of a vector of nodes and a vector of links. The vector of nodes contains the information about the index of the node, for example, node No. 0 means that the node is the depot. The vector of links contains the links pertinent information that include start node, end node, start service time, end service time, cost, travel time, number of times of service, etc. In network initialization, we also maintain the information regarding the shortest paths and distances between any pair of nodes. Since we use the shortest path and distance frequently, when we initialize the network we use Floyd's shortest path algorithm (Floyd, 1962) to generate all pair-wise shortest paths and distances and save the information for later use.

3.2.3 Operations

There are several operations that are performed during the implementation of the algorithm. These operations are frequently used in the algorithm and are as follows.

3.2.3.1 Finding the Shortest Paths and Distances for All Node Pairs

Floyd's shortest path algorithm (Floyd, 1962) is used to generate all pair-wise shortest paths and distances between any pair of nodes in the network. We will frequently use the shortest path between two nodes.

3.2.3.2 Finding the Furthest Link from the Depot

For all existing links, the shortest distance between the start nodes of those links to the depot, and the shortest distance to the end nodes of those links are found. These distances are added together and compared across all links. The link that has the highest sum of these two numbers is the link that is the furthest from the depot.

3.2.3.3 Finding the Nearest Link to a Specific Route

When we attempt to insert a link in a route, we always try to insert links that are close to the route. The procedure to find the nearest link is slightly complicated. A link is deemed nearest to a route if either of the following two conditions is met. If all the links in the route are serviced, then the nearest link is the one with the smallest distance from any node on the route to that link. If some links on the route are not serviced, then we delete the links that are not serviced and insert the link and compute the saving as the inserting cost. The procedure is as follows:

Given a link, and a route

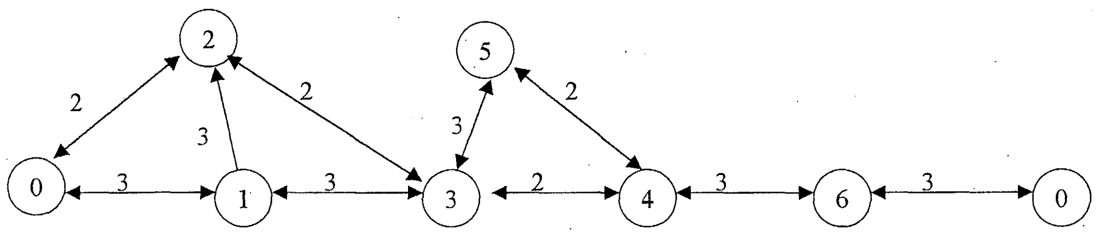
Step 1. Compute the shortest distances from every node in the route to the start node of the link and select the smallest one.

Step 2. Compute the shortest distances from the end node to every node in the route select the smallest one.

Step 3. Sum the total distance.

Step 4. Compare all the links that need to be compared, the link with the smallest total distance is the nearest link to the route.

Assume for example that we have three existing links (0-1, 2-3, 3-5) and a route (0 1 3 4 6 0), as shown in the next network:



The smallest distances from the nodes in the route to the start nodes of the links are:

0-0	0
0-2	2
3-3	0

Assume the smallest distances from the end node to every node in the route are:

1-1	0
3-3	0
5-4	2

So the total costs are:

0-1	0
2-3	2

3-4 2

So the nearest link to the route is 0-1 which is actually on the route.

3.2.3.4 Augment Operation

The augment operation is trying to see whether one route can service another route, this operation is suggested by Golden (1983). The only difference is that we should consider that the links are directed.

3.2.3.5 Merge operation

The merge operation also borrows the idea of Golden (1983). Since his algorithm deals with the non-directed case, it is modified to account for directed network. The steps are as follows:

Step 1. Find all common nodes of the two routes.

Step 2. Exchange the routes at the point of the common node and generate the new route that provides the largest saving and can service the two original routes. The generated route should respect the direction of the two original routes.

Step 3. Loop the procedure until all common nodes are checked.

Step 4. Select the largest saving exchange point and merge the two routes.

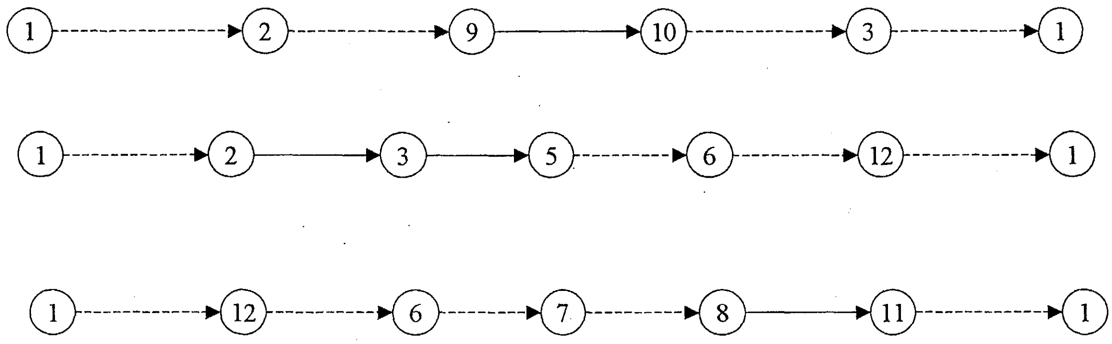
The main difference between the proposed algorithm and Golden's algorithm is that this algorithm would keep the direction of the route, because the network is a directed network.

To show the difference, we use the same example from Golden (1983).

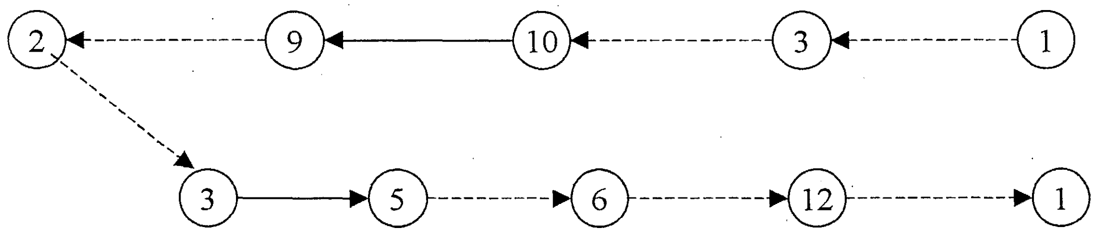
Route 1: 1 2 <u>9 10</u> 3 1

Route 2: 1 <u>2 3 5</u> 6 12 1

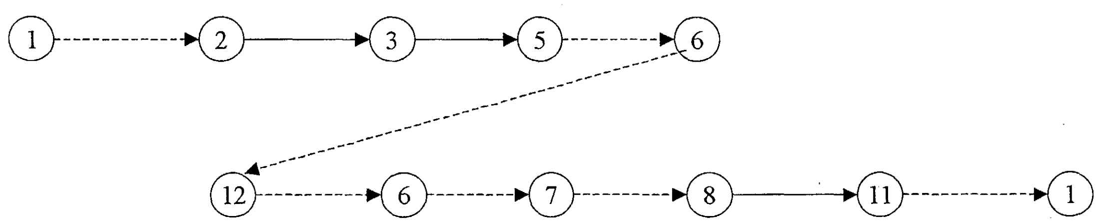
Route 3: 1 12 6 7 <u>8 11</u> 1



In the non-directed network, route 1 and route 2 can be merged into 1 3 10 9 2 3 5 6 12 1.



However, in the directed network, this route is not correct, because we should keep the direction. Note that the non-directed network can provide a lower bound for the directed network. In the directed network, routes 2 and 3 can be merged into 1 2 3 5 6 12 6 7 8 11 1. This route is feasible in the directed network.



This is a simple procedure that produces good solutions for small networks very quickly. When applied to a real world problem, this procedure also produces very good results. Even for small networks, the algorithm proposed in this research can enhance the solution that is obtained from the merge algorithm. For the larger real world problems, as the computation results show, we see that just using the merge algorithm is not efficient and we should find some other improvement algorithms.

3.2.3.6 Delete operation

This operation tries to delete one serviced link of a specific route. When we delete a link that is serviced by a vehicle, we also should delete the neighboring links that are not serviced by the route. This is because these links only serve as means of connection for the link that is serviced in the vehicle's route. When the serviced link is deleted, there is no need to traverse the links that are not serviced. The delete operation works according to the following steps.

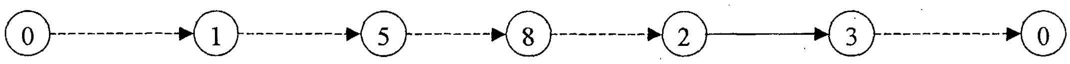
Step 1. Delete all links that are next to the link that is to be deleted and are not serviced by the route.

Step 2. Repeat the operation until the links that are serviced by the route are found.

Step 3. Reestablish the route.

As an example, consider the route 0 1 5 8 2 3 0 in which links 5-8 and 2-3 are serviced. Assume that we want to delete link 5-8. After deleting link 5-8, we need to delete link 1-5 and 0-1 because these links are not serviced as part of the route. The only remaining link to be serviced is 2-3. If the shortest path from depot to node 2 is link 0-2, then the reestablished route after deleting link 5-8 will be 0 2 3 0. The operation is show as follows:

Before deleting link 5-



After deleting link 5-

The delete operation in this case is different from the delete operation in the VRP. In VRP the operation is deleting nodes, therefore, after deleting the nodes we just need to re-link the route. But in the arc routing problem, we have to delete links. A simple re-linking will result in the same route as before, therefore, the operation is more complicated than the delete operation in the VRP. The programming experience also shows that programming the delete operation in the arc routing problem is more difficult than programming the operation for the VRP.

3.2.3.7 Inserting Operation

Inserting a link in an existing route is not as easy as it is in the VRP. We not only should perform a capacity check, but also we should find where is the best position to insert the link in the route. The following steps accomplish the inserting operation.

Step 1. If the link demand plus the total route demand is greater than the vehicle capacity then set the least insert cost as a large number (capacity check).

Step 2. If the sum is less than the vehicle capacity and not all the links in the route are serviced, find the start node and end node for each continuing links that are not serviced by this route. If the all the links in the route are serviced go to step 5.

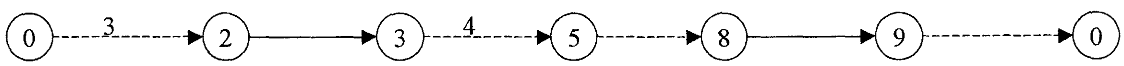
Step 3. Find the sum of the shortest distances between the start node in the route to the start node of this link and the end node of this link to the end node in the route. The inserting cost is the cost of the after-inserting route minus the cost of before-inserting route.

As an example, consider the route 0 2 5 8 9 0. Link 5-8 is the only service link in this route. Now consider link 2-3 for insertion. The shortest distance from 0 to 2 is 3 and from 3 to 5 is 4. The distance 0 2 5 is 5. After inserting link 2-3, the route will be 0 2 3 5 8 9 0, and the vehicle will service link 2-3 and link 5-8.

Therefore, the insertion cost is 3+4-5 = 2. The procedure is shown in following diagram.

Before inserting link 2-3

After inserting link 2-3



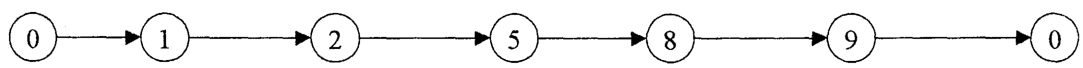
We also can insert this link between nodes 8 and 0 that results in route 0 1 2 5 8 2 3 0. Of course we need to calculate the insertion cost for this operation as well.

Step 4. Check all possible insertion position and choose the position that has the least insertion cost. Insert the link at that position.

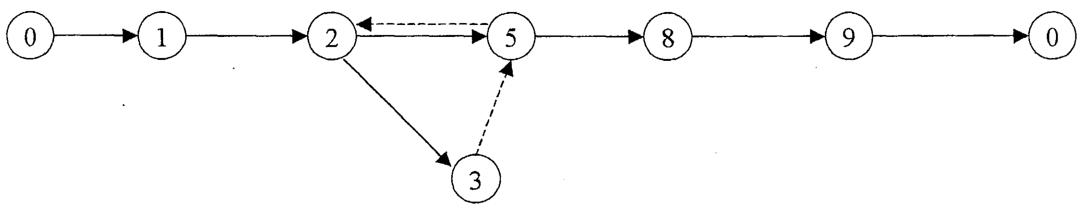
Step 5. If every link in the route is being serviced, we can find the nearest node in the route to the link that is under consideration for insertion. The nearest node can be found by the computing the shortest distance from the node (whether it is starting node or ending node does not matter) in the route to start node of the link and the end node of the link to the node in the route.

As an example, consider the route 0 1 2 5 8 9 0. All links are being serviced in this route and we are considering inserting link 2-3 in the route. The nearest node is 5, and the final route is 0 1 2 5 2 3 5 8 9 0. The algorithm allows a loop in the route.

Before inserting link 2-3



After inserting link 2-3



The deletion and insertion operations allow us to exchange links between two routes. These two operations are the main operations based on which the improvement algorithm is designed.

3.2.3.8 Conclusions

The operations discussed above are basic operations in the arc routing problems. Many ideas are borrowed from the vehicle routing problem. However, the VRP operations cannot be directly applied in the arc routing problem and some modifications are necessary to make the operations applicable to the ARP. Since we are dealing with directed networks, we have to pay special attention to ensure that the route directions are maintained. This makes the programming a little more difficult. These operations can be combined together to generate different algorithms. Other operations can also be developed to enhance the procedures and algorithms just like those in the VRP.

3.2.4 Initial Solution

In this research, two algorithms are used to generate an initial solution. Since we apply an improvement heuristic to improve the initial solution, the first algorithm to generate an initial solution is quite simple. The second one is a little more complicated but it will result in a relatively good initial solution. We use two techniques to generate an initial solution so as to be able to assess the impact of the initial solutions on the final results.

The first algorithm works as follows. For each link that requires service, use one vehicle to service the link and find the shortest path from the depot to the start node of the link and from the end node of the link to the depot. A route is then established for each link following this shortest path. This is a very simple procedure that can construct an initial feasible solution very quickly.

The second procedure is slightly more complicated. It attempts to find the furthest link from the depot, and use the shortest path from the depot to the start node of that link and from the end node of that link to the depot to establish a route. When the route is established the procedure attempts to insert the nearest links in this route. The steps of the procedure are as follows:

Step 1. For existing links that are not serviced, find the furthest link to the depot using the operation discussed in Section 4.3.2.

Step 2. Find the shortest path from the depot to the start node of this link and from the end node of this link to the depot. Establish an initial route using the links in the shortest path and the link found in step 1.

Step 3. Use the operation discussed in Section 4.3.3 to find the nearest link to the current route.

Step 4. Insert the link found in step 3 in the route.

Step 5. Loop until the capacity constraint for vehicle is violated.

Step 6. Loop until no link that requires service remains.

The basic idea here is simple, we always try to service the links that are close to each other. This algorithm can provide good initial solution, but it is more tedious. We have proposed two algorithms to obtain an initial solution thus far. Next we will propose a merge algorithm that is a revised version of that proposed by Golden (1983), and a newly developed improvement algorithm.

3.2.5 The Merge Algorithm

Golden's merge algorithm (1983) can be viewed as an improvement algorithm. Comparing with the exchange algorithm in the VRP, we find that the merge algorithm is very similar to the exchange algorithm. The procedure is as follows:

Step 1. Select two routes.

Step 2. Perform merge operation on these two routes

Step 3. Loop over all pair-wise routes, until no improvement can be found.

The Merge algorithm is very efficient for small networks. It can find a good solution very quickly. Its limitation is that it tries to combine two routes into one. Therefore, if the capacity of the vehicle does not allow servicing the two routes, the algorithm will discard the possibility of exchanging two links between the two routes and hence improving the solution.

3.2.6 The Augment Algorithm

Golden's Augment algorithm (1983) can also be viewed as an improvement algorithm. The advantage of this algorithm is that it can reduce the number of routes quickly, we will discuss this later in later Sections. The procedure is as follows:

Step 1. Select two routes.

Step 2. Perform merge operation on these two routes

Step 3. Loop over all pair-wise routes, until no improvement can be found.

3.2.7 Link Exchange Improvement Algorithm

Most algorithms for solving the arc routing problem just try to construct a solution. Though many scholars have suggested using some sort of algorithm to improve the solution, no implementations of such algorithms were found in the literature. In this research we will apply an improvement algorithm to improve the solution generated by the modified merge algorithm of Golden.

The procedure borrows the idea from the link exchange heuristics for the VRP as most researchers have suggested. Actually, if we consider the merge algorithm carefully, we find that the merge algorithm is very similar to the 2-opt algorithm (Potvin and Rousseau, 1995) or Or-opt algorithm (Or, 1976) in the VRP. The improvement algorithm that we propose here is similar to the cross operation in the VRP, but it is much more complicated than simple crossing.

The improvement algorithm is as follows.

Step 1. Select two routes.

Step 2. Delete one link from each route, and insert the link into the other route.

Step 3. Compute the saving (the cost difference between before and after deleting and inserting) for each pair of links.

Step 4. Select the largest saving link pair as the next neighbors.

Step 5. Check all routes until no improvements can be made.

The constraint of the algorithm comes from that it must exchange the links, so the number of link that are serviced are the same for each route. Although the merge algorithm can merge the two routes into one route, it requires that the new route must service all the links in the two original routes. We propose another algorithm to solve this problem.

3.2.8 Delete and Insert Algorithm

This procedure borrows the idea from the Or-opt heuristics for the VRP as most researchers have suggested. The algorithm tries to delete one link in one route and insert the deleted link into another route. The procedure attempts to find the largest saving and insert the link into the route that results in the largest saving. Delete and insert algorithm can change the number of the links serviced by the route and does not require the new route to service both original routes.

The improvement algorithm is as follows.

Step 1. Select two routes.

Step 2. Delete one link in one route, and insert the link into the other route.

Step 3. Compute the saving (the cost difference between before and after deleting and inserting) for each operation.

Step 4. Select the largest saving operation and solution as the next neighbor.

Step 5. Check all routes until no improvements can be made.

The limitation for this algorithm is that it just deletes one link and inserts one link. There are possibilities that we can delete more than one link from a route and insert more than one link to another route or routes. Investigating these possibilities is left for future research.

3.3 Conclusions

In this Section we discussed several operations that are performed on directed networks. These operations are very basic for network optimization and especially arc routing problems. Several improvement algorithms were suggested to improve the initial solution. As was discussed before, every algorithm has its limitations. To avoid the local optima caused by specific operations, it is a good idea to combine these algorithms together. However, one should note that different combinations also have different limitations (being trapped in the vicinity of an inferior solution). Therefore, the problem becomes how to combine these algorithms together.

Many meta heuristics have been suggested. These meta heuristics can be applied in several areas. Some researchers also suggested using these meta heuristics in arc routing problem (Win, 1987; Jin Yuan Wang and Jeff R. Wright, 1994), however, there are no such results reported up to now. It is believed that there are no such basic operations implemented in arc routing problems because of the special characteristics of the arc routing problems. Computational experiences also show that programming of algorithms for arc routing problem is more difficult than the node routing problems.

In Section 4, we will discuss the application of the proposed algorithms for solving a real-world problem. We will also apply the proposed algorithms in the next Section to several test problems.

4. Model Implementation for Designing Calvert County Salting Route

4.1 Background

In Calvert County the salting routes are divided into four sections. Lenhart (1998) studied the existing snow emergency route structure in Section I of Calvert County. His study was exploratory in nature. He presented a preliminary mathematical formulation for optimizing the route structure and solved problem exactly. The result was a significant improvement over the original route design greatly.

Unfortunately, his approach is not practical for solving the problem for larger sections. The computational time required to solve the problem exactly is prohibitive. To solve a very small problem exactly using Lenhart's approach takes over 10 hours on a Pentium 200 machine. Lenhart's study is valuable because it shows that there is possible room for improvement in the snow emergency route design in Calvert County.

In this research we will consider the routes in all four sections of the salting network. We use the algorithm proposed in this research to design the route structure for these sections and compare our results with the current route structure that is in place in the county.

4.2 The Algorithm

Due to the nature of the problem an important aspect should be accounted for very carefully in any algorithm design. This aspect of the problem is that both lanes of a two-lane highway can be serviced by a truck that is traveling in one direction only. So we can strategically insert the link into an existing route to see which direction has the least inserting cost. First when we try to create an initial the solution, we do not consider servicing the two-lane highways. After we obtained an initial solution, we attempt to insert the two-lane highway links in each direction into the current solution and choose the direction that has the least insertion cost in the initial solution.

We can also initially select the direction randomly and try to find a better solution later. The problem for this method is that we cannot change the service direction at runtime. Another method is that at runtime, we randomly change the service direction, the problem in this case is that there is no guarantee that the solution is still feasible. The procedure we use to obtain the solution for the case study is summarized below.

<u>Step 1</u>. Initialization

Use the second initialization method to generate the initial solution and assign the tandem trucks to the longest routes.

<u>Step 2</u>. First Improvement

Apply the augment algorithm, merge algorithm, and delete and insert algorithm successively. We do not want to use the exchange algorithm here because we want to reduce the number of the routes first.

<u>Step 3</u>. Second Improvement

Apply the merge algorithm and delete and insert algorithm again and then use the exchange algorithm. We do not use the augment algorithm in this step because it is almost impossible to get any improvements from the augment algorithm.

<u>Step 4</u>. Third Improvement

Continue with the application of the delete and insert algorithm and the exchange algorithm until no more improvements can be made. We do not use the merge algorithm any more because it can hardly improve the solution.

The final solution is obtained when the steps above are sequentially applied and we cannot make any more improvements with the application of Step 4.

4.3 Application in Calvert County

The above procedure was applied to the existing network of salting operations in Calvert County. We applied the procedure in three different scenarios. These scenarios involved designing salting routes for Sections I and II, Sections III and IV, and all four sections combined. In these applications, the service miles were limited to 16, 18, and 20 miles while no limitation was imposed on the total route lengths for the trucks. 16 maximum service miles is selected to indicate the current salt usage. 18 and 20 maximum service miles indicate slightly less slat usage. In each scenario we combined the salting network sections to generate a relatively large network for application of the proposed algorithm to optimize the assignment.

4.3.1 Current Route Assignments

Currently, there are fourteen trucks servicing this area and all of them have the same salt carrying capacity of 16,000 lbs (8 tons). Figures 4.1 through 4.14 and Table 4.1 show the current route assignments statistics for each of the trucks that are identified by their number.

The total travel distance is 400.70 miles and the deadhead distance is 243.71 miles. The maximum and minimum route lengths are 53.12 and 15.46 miles respectively. The average route length and the average deadhead mileage are 28.62 and 17.41 respectively. This route structure is designed by the maintenance personnel based on their many years of experience.

4.3.2 System Route Assignments

The proposed procedure was used to develop new route assignments. The results of the application of the procedure are promising. The obtained solution not only saves deadhead miles, but also reduces the number of used truck.

When a service mile constraint of 16 miles is imposed, in Sections 1 & 2, the number of trucks used is reduced by one, however, the total route length and the total deadhead miles is increased by 8.75% and 13.12% respectively compared to the existing service. In Sections 3 & 4, the number of used trucks is reduced by one while at the same time the total route length and the total deadhead miles are also reduced by 11.30% and 12.14% respectively. When all sections are combined, the number of used trucks is reduced by 2 and at the same time, the total route length and the total deadhead miles are reduced by 5.28% and 11.81% respectively.

When the limit on maximum service miles is increased, the savings on the number of trucks, the total route length, and the total deadhead miles are more pronounced. These savings range between 10.39% to 17.66% in total route length, 18.36% to 32.16% in total deadhead miles. These savings can be realized while the number of used trucks can also be reduced (up to 5).

One should note that the basis for comparison between the routes proposed by the system and the current routes should only be the proposed system routes when the 16 mile service limit constraint is imposed. The reason for this is that according to maintenance personnel, the maintenance office currently uses 500 lbs of salt per lane mile in an average storm. Using this number and given the capacity of the trucks used, the maximum service mileage for a truck is 16 miles. The research team provided the results of the other runs with maximum service miles constraints of 18 and 20 as a comparison base so that the potential for savings in deadhead miles and number of trucks can be realized if the salt utilization can be slightly reduced.

One should also note that the proposed algorithm is still in its preliminary stages. As is, the algorithm does not consider some of the practical operational constraints. One important constraint that is ignored is the route continuity constraint. It is desirable that the routes that are assigned to trucks be continuous. This is important in practice to avoid driver confusion with respect to his or her service route. This feature is not present in the current algorithm but must be considered in a final version that is to be implemented. The research team is currently working on this and other extensions to the algorithm. It is expected that the implementation of this constraint will reduce the savings in the deadhead miles due to the more constrained nature of the problem.

Overall, these results indicate that there is potential for savings if a full-blown decision support system is in place to assist the maintenance managers in making these route assignment decisions.

The route assignments developed by the system with the service route length constraint of 16 miles for Sections I & II, Sections III & IV, and all Sections are shown in Figures 4.15 through 4.38. All Figures are properly annotated to indicate which scenario they are representing and what constraints are included in their development. Tables 4.2 through 4.4 summarize the detailed results for all scenarios considered. Tables 4.5 through 4.7 provide a comparison of the results obtained from the system implementation and the current service. In these tables, the blue and red prints represent the percentage decrease and increase from the current route lengths resulting from the implementation of the corresponding system routes respectively. As these results indicate, in all cases, the routes proposed by the system improve upon the current routes.

5. Conclusions and Directions for Future Research

In this research we proposed an approach for a real-world snow emergency route assignment. The proposed approach seems to be very efficient in solving problems of the size equivalent to those in Calvert County. The final solutions were obtained in less than 15 minutes on a Pentium 200 with 266 MHZ clock speed. This indicates that the algorithm can easily be implemented in a decision support system that can be used to examine what-if scenarios.

Another important aspect is that the algorithm can do the route assignment dynamically. The salting consumption rate varies with the storm characteristics and the current snow route assignment is designed using an average consumption rate. Using this procedure, one can develop different plans based on different levels of salt consumption rates.

This study indicates that the proposed procedure can be applied to real-world problems successfully. Although the problem is in a relatively small network, we still are able to have improvements over the original route design.

As an immediate and most important extension to the current system we need to develop a graphical user interface for the current model. An interface that allows the users to evaluate the what-if scenarios by just looking at the details of the service networks they generate on a computer screen is essential to the successful implementation of the prototype system in actual practice. Such an interface can be designed to clearly identify the service and the deadhead travel and help the user identify the implications of the changes that he or she makes and evaluate the impacts of various operational constraints.

A second step in extensions of the current system is to implement it in another County that has a more sophisticated network. To that end certain modifications to the underlying model and algorithms would no doubt be necessary, however, the effort invested in making these modifications will have significant return in terms of operational cost savings for the State of Maryland.

Because the particular nature of the snow emergency route assignment problem, the main issues that need to be investigated further in the future for implementation in other counties are as follows.

As we discussed in the previous Sections, for two-lane highways both directions can be serviced at one time. We should find in which direction the two-lane highways should be serviced. Theoretically, there are two ways to change the service directions. First we can change the service direction at the initial stage and generate an initial solution. The second is changing the service direction at the improvement stage. When we improve the solution we can try to change the direction of service to see whether we can improve the solution even further. The first one is time consuming but can avoid getting trapped at inferior solutions. The second one is fast, but it is possible that it will result in convergence to an inferior solution.

Another problem that needs further attention is the truck capacities. The main issue is which type of truck must be used for what type of route. We find that the final solution is highly sensitive to the choice of the initial solution. Our algorithm cannot automatically deal with this. More advance algorithms are needed to deal with this problem. This is left for future research.

The problem, which was dealt with in this research, was just a single depot problem. We only dealt with the Sections of the Calvert County that were services from a single salt dome. In general there may be several salt domes at various location in the service area. Therefore, the problem becomes a multiple depot problem arc routing problem. More sophisticated algorithms need be developed to deal with this problem. This is left for future research.

Table 4.1

Current Route Assignment Statistics

Sections 1&2 (Current Service)

Section No.	Truck No.	Total	Service	Deadhead
1	053	22.42	9.67	12.75
1	106	25.56	7.56	18.00
1	114	26.46	11.06	15.4
1	323	31.17	9.08	22.09
2	057	36.18	14.62	21.56
2	315	30.84	15.70	15.14
Total		172.63	67.69	104.94
Average		28.77	11.28	17.49

Sections 3&4 (Current Service)

Section No.	Truck No.	Total	Service	Deadhead
3	034	17.45	9.11	8.34
3	103	15.46	11.67	3.79
3	109	15.60	11.27	4.33
3	283	17.86	9.98	7.88
3	324	31.50	13.36	18.14
4	108	42.02	3.49	38.54
4	111	35.06	17.20	17.86
4	281	53.12	13.23	39.89
Total		228.07	89.31	138.77
Average		28.51	11.16	17.35

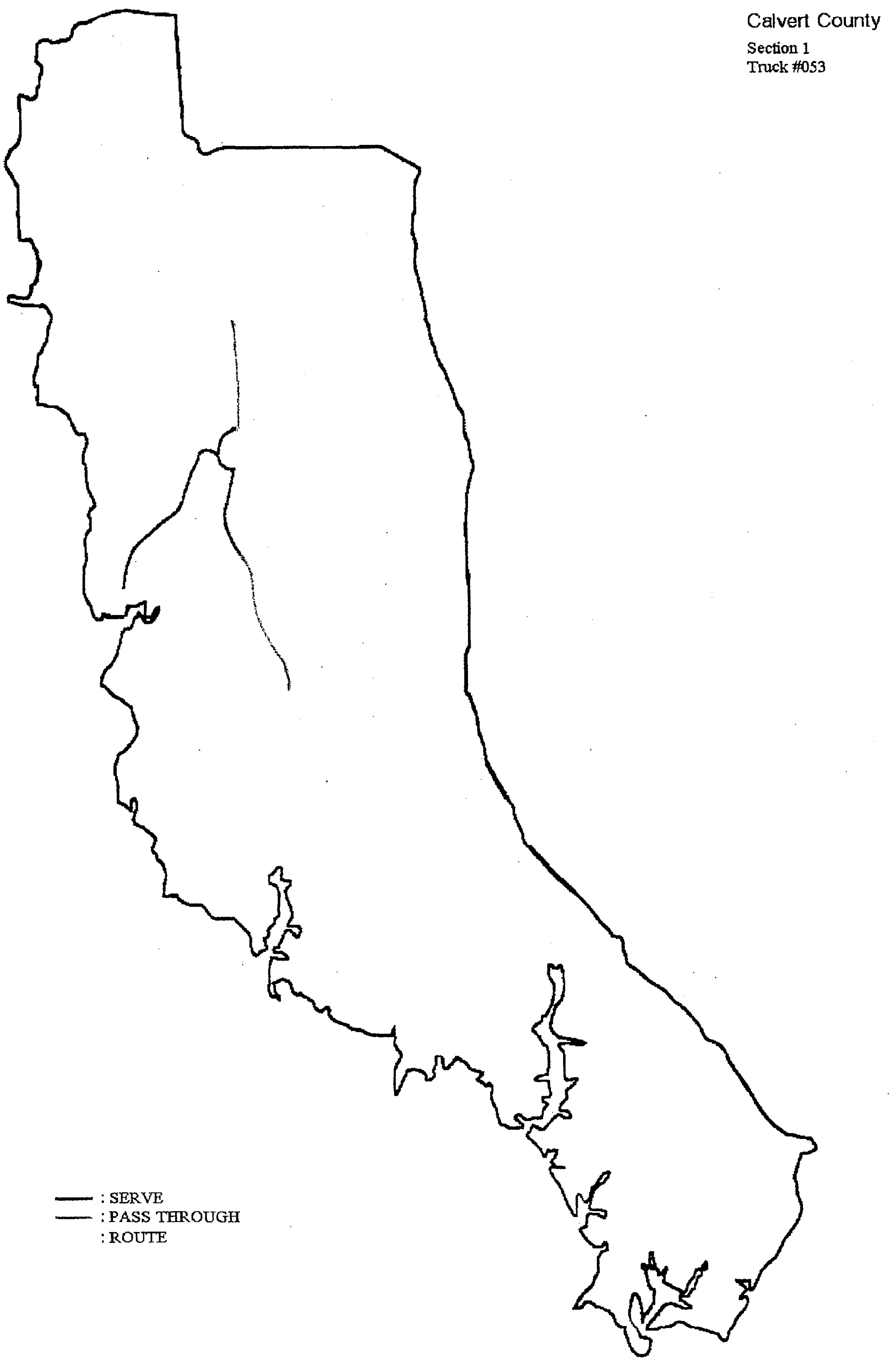


Figure 4.1 Calvert County Current Routes, Section I, Truck 053

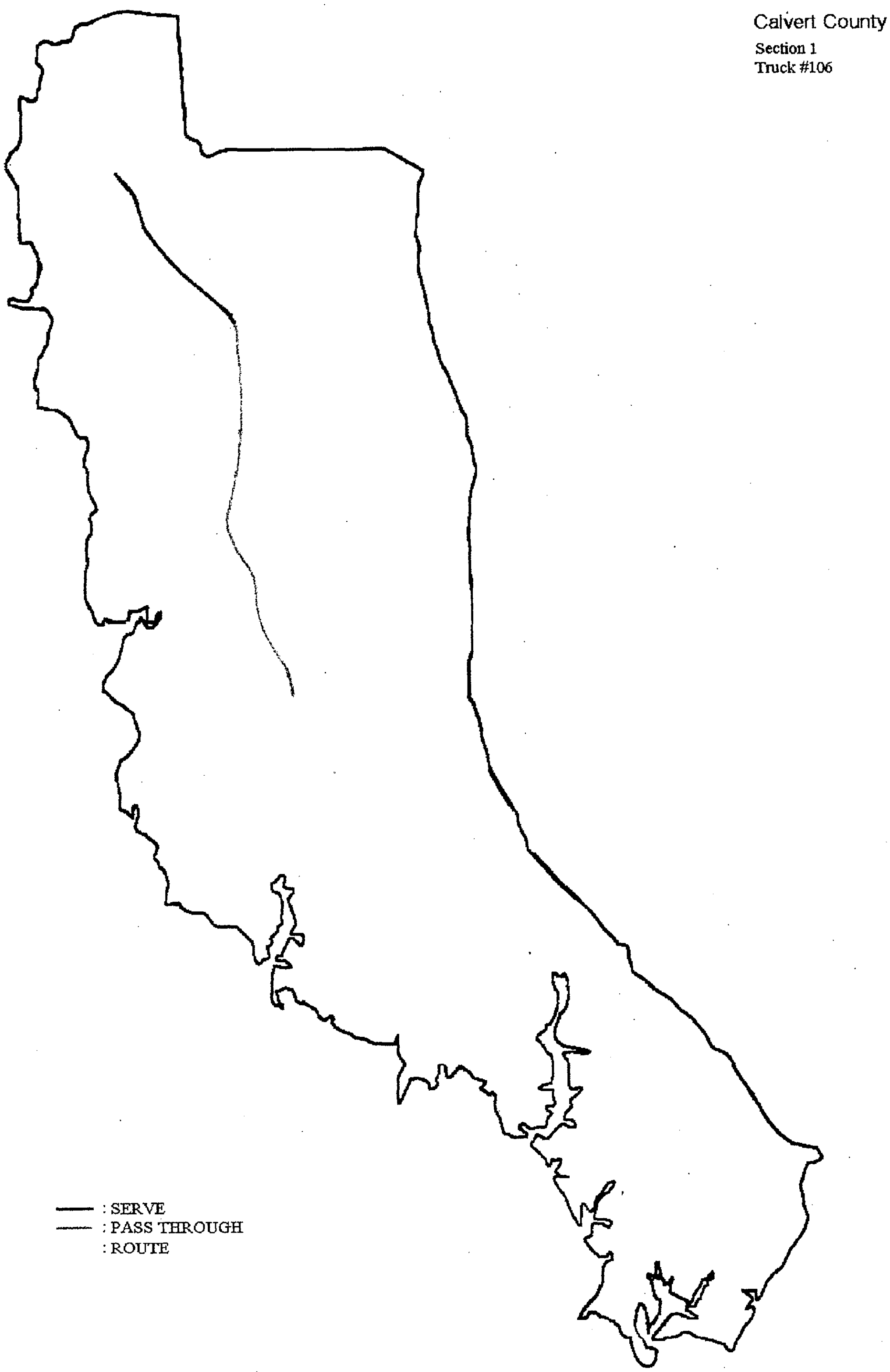


Figure 4.2 Calvert County Current Routes, Section I, Truck 106

Figure 4.3 Calvert County Current Routes, Section I, Truck 114

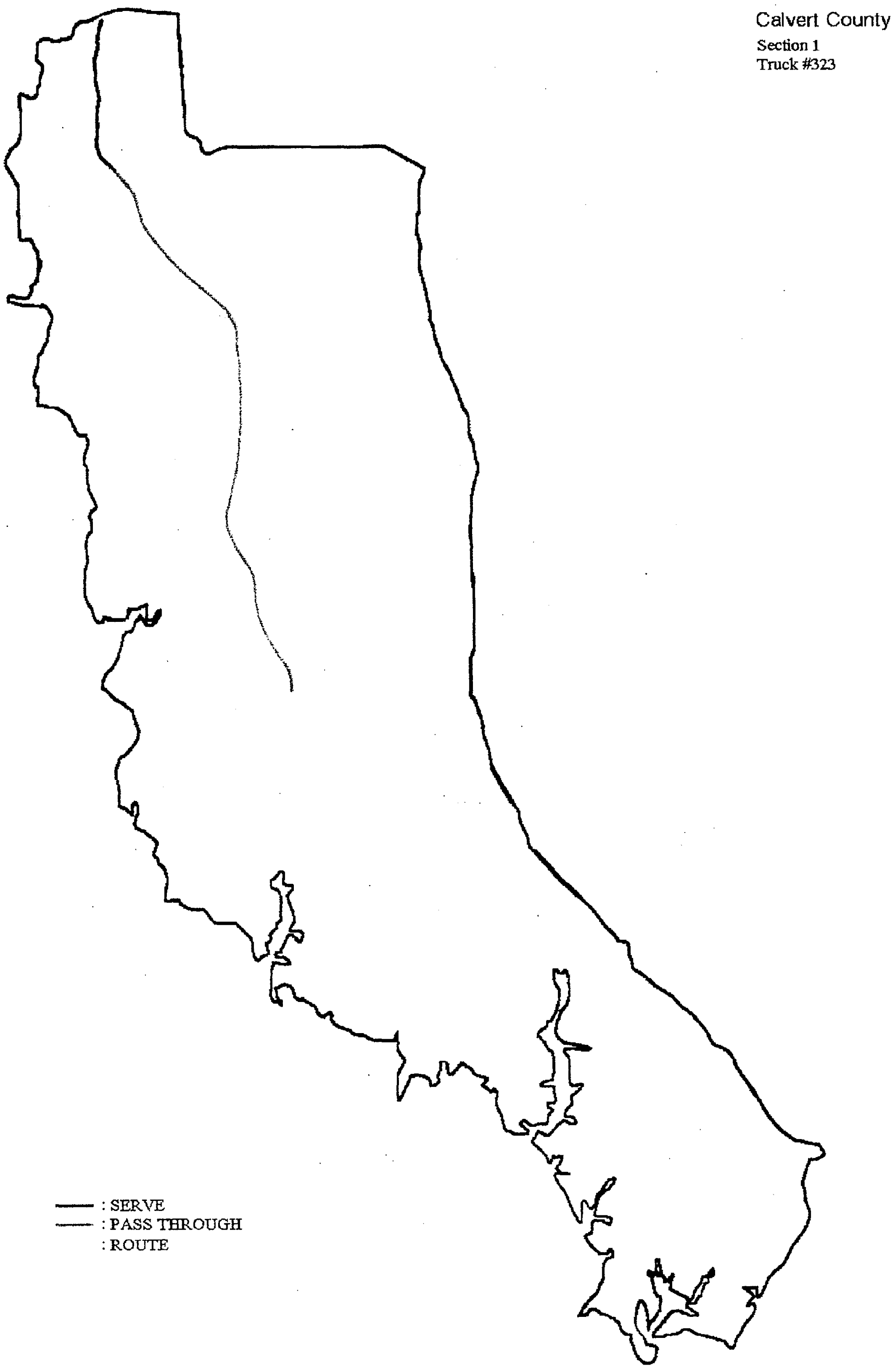


Figure 4.4 Calvert County Current Routes, Section I, Truck 323

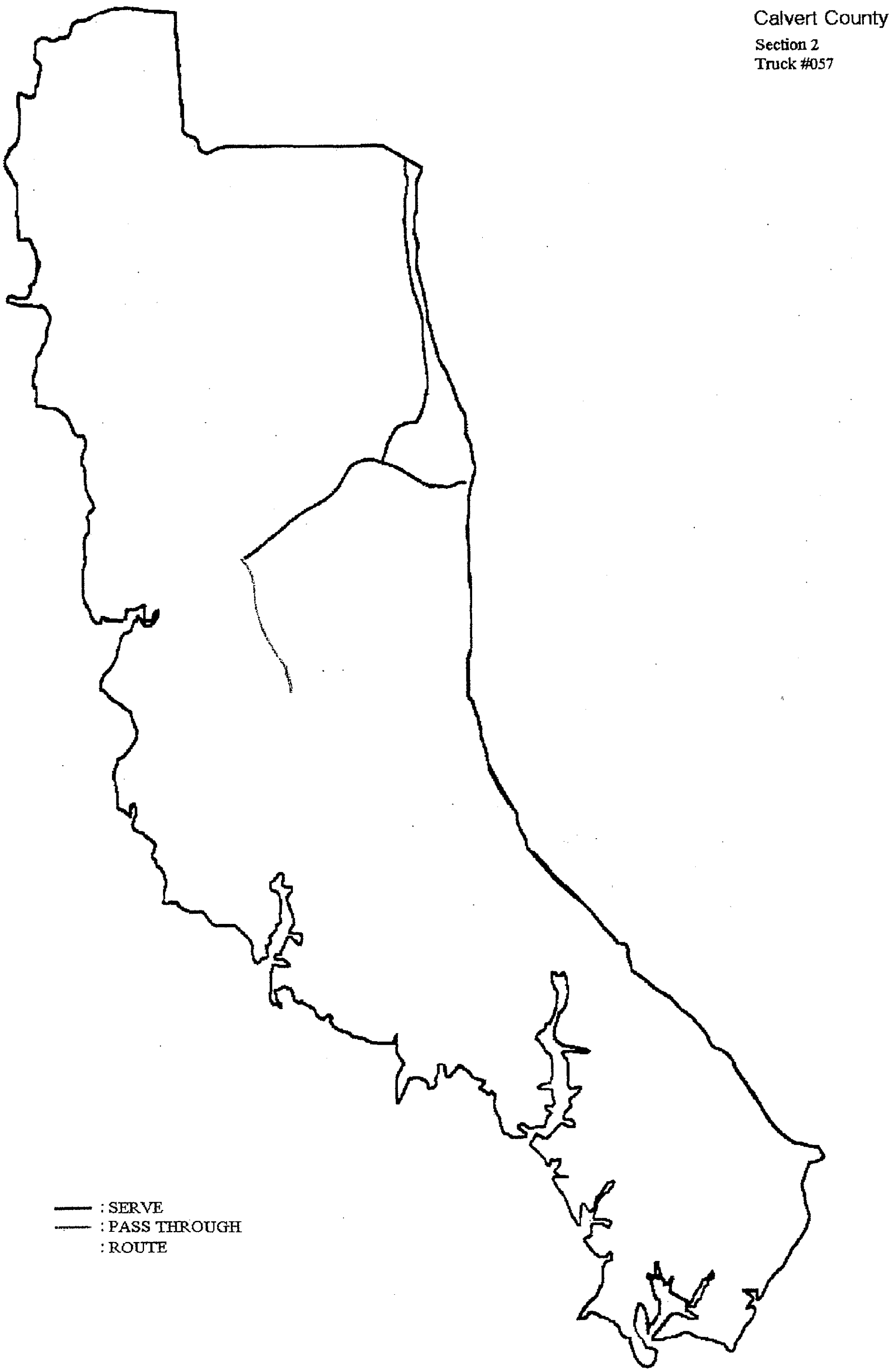


Figure 4.5 Calvert County Current Routes, Section II, Truck 057

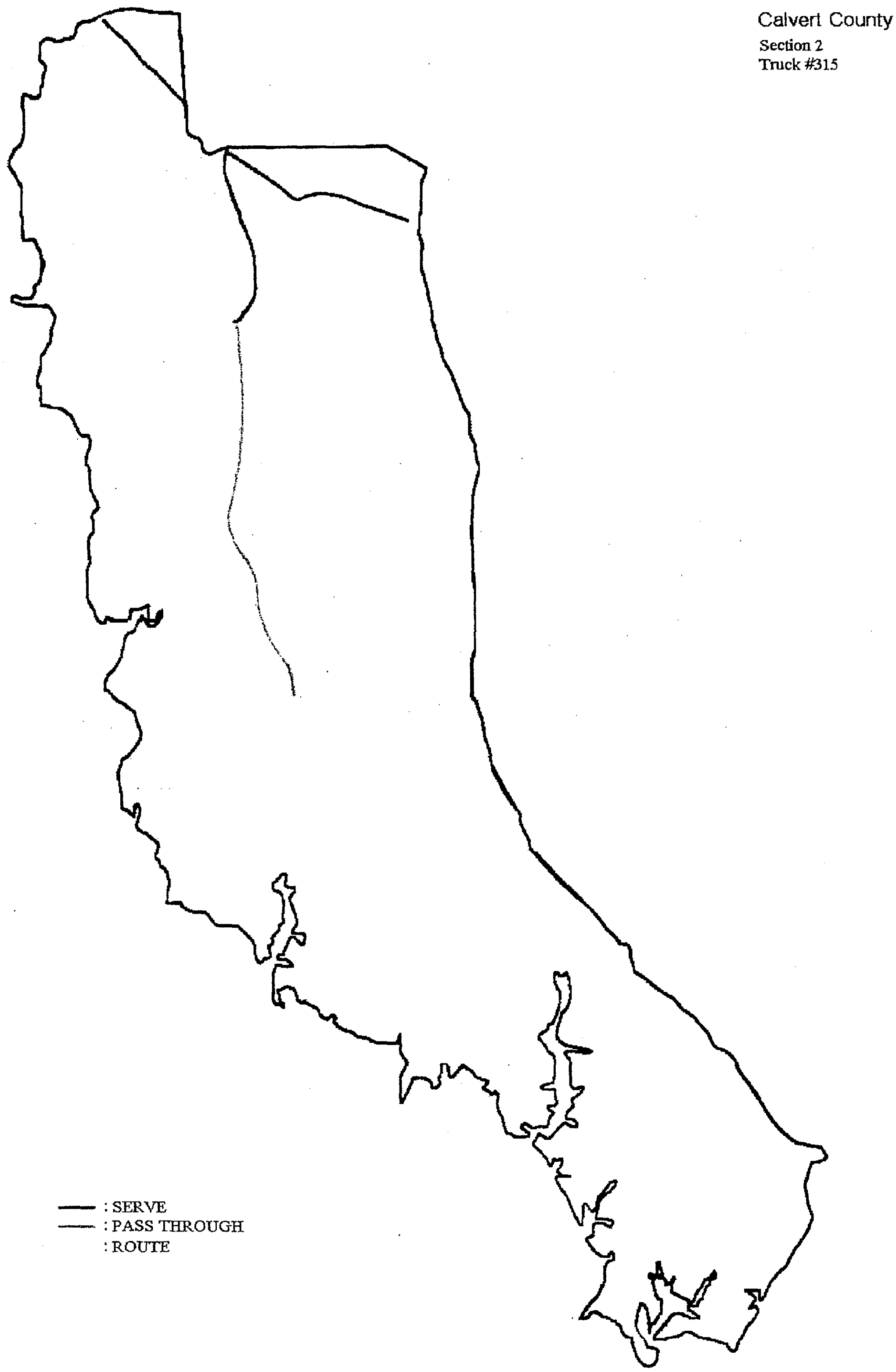


Figure 4.6 Calvert County Current Routes, Section II, Truck 315

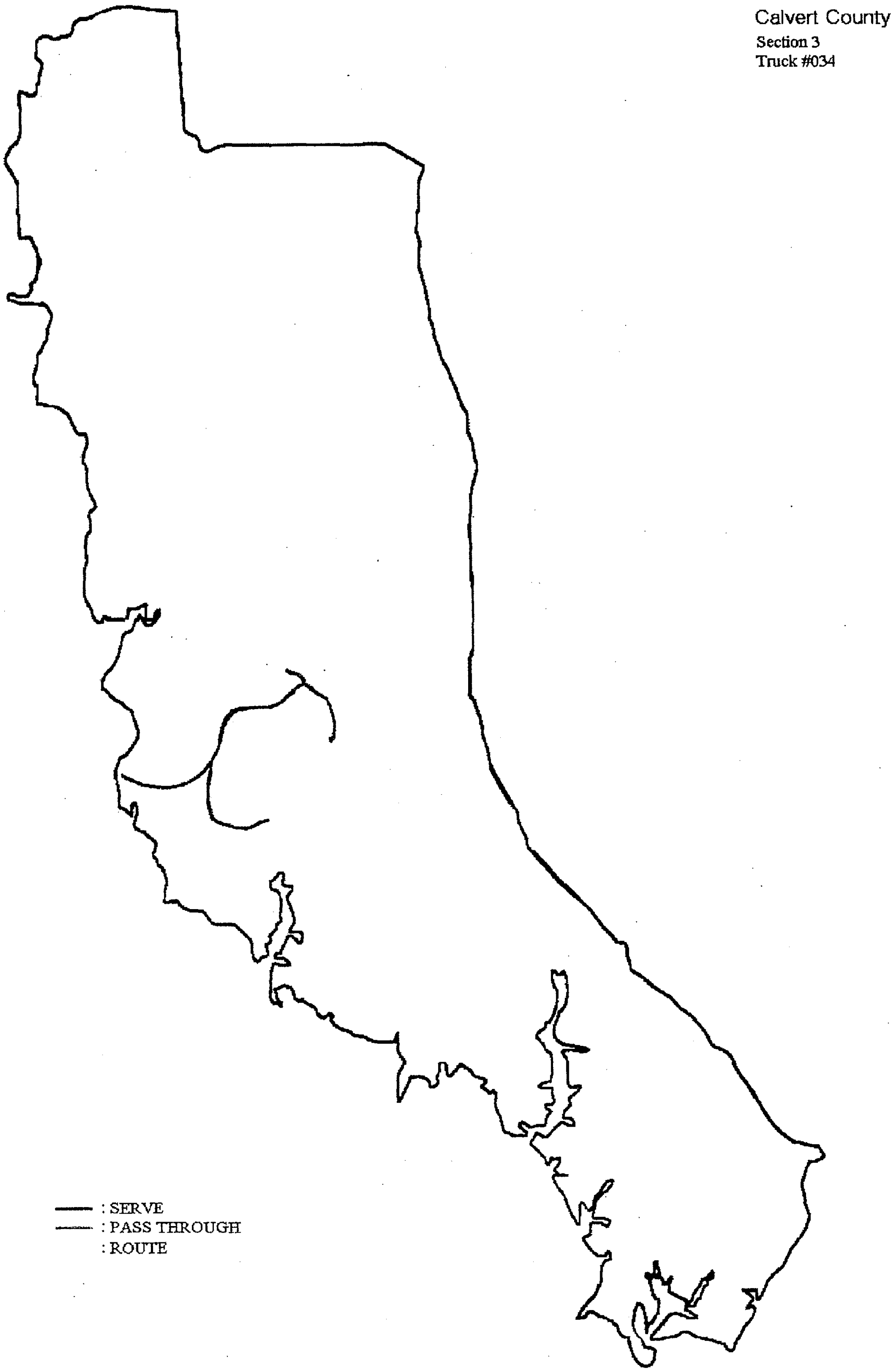


Figure 4.7 Calvert County Current Routes, Section III, Truck 034

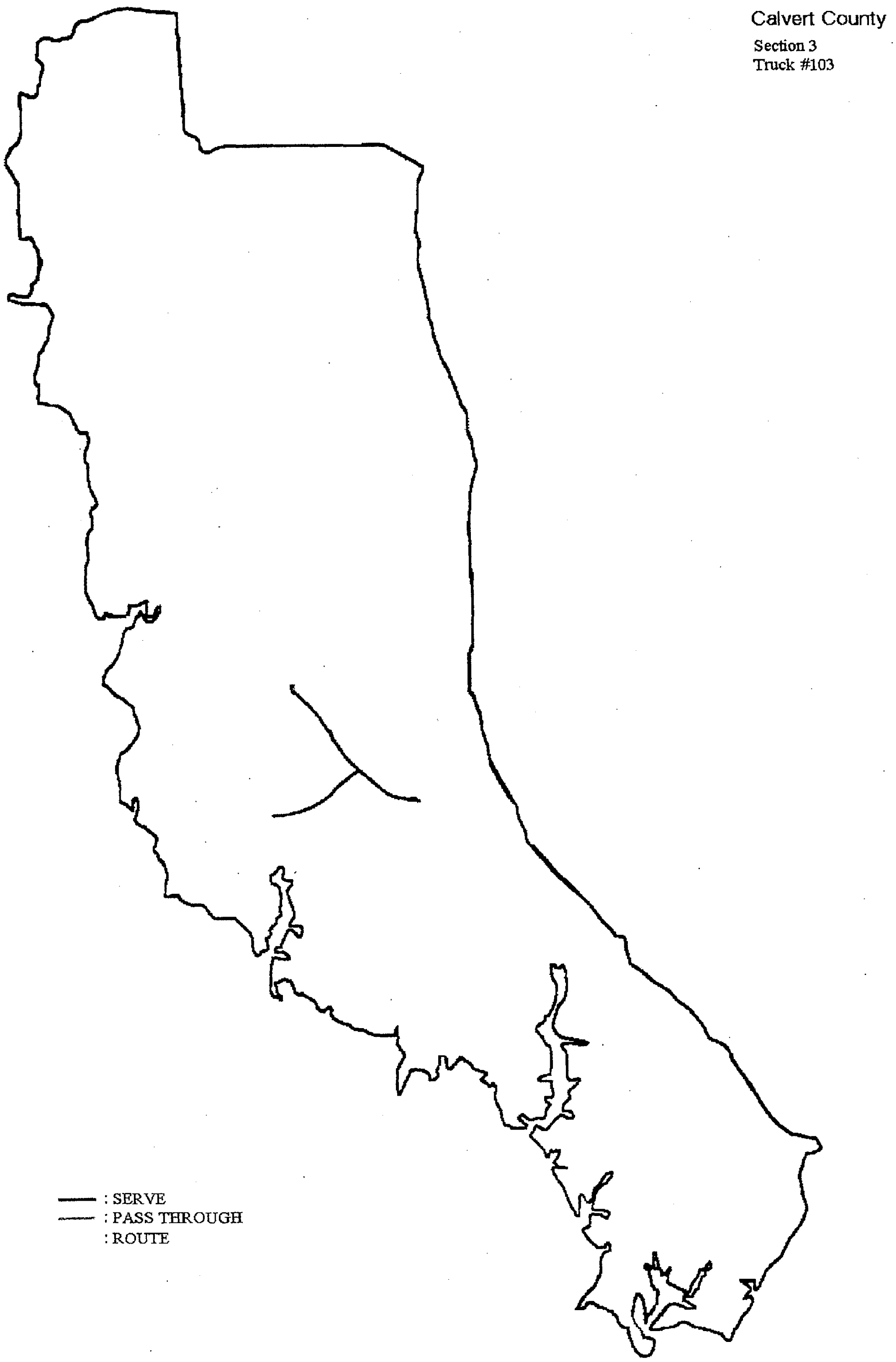


Figure 4.8 Calvert County Current Routes, Section III, Truck 103

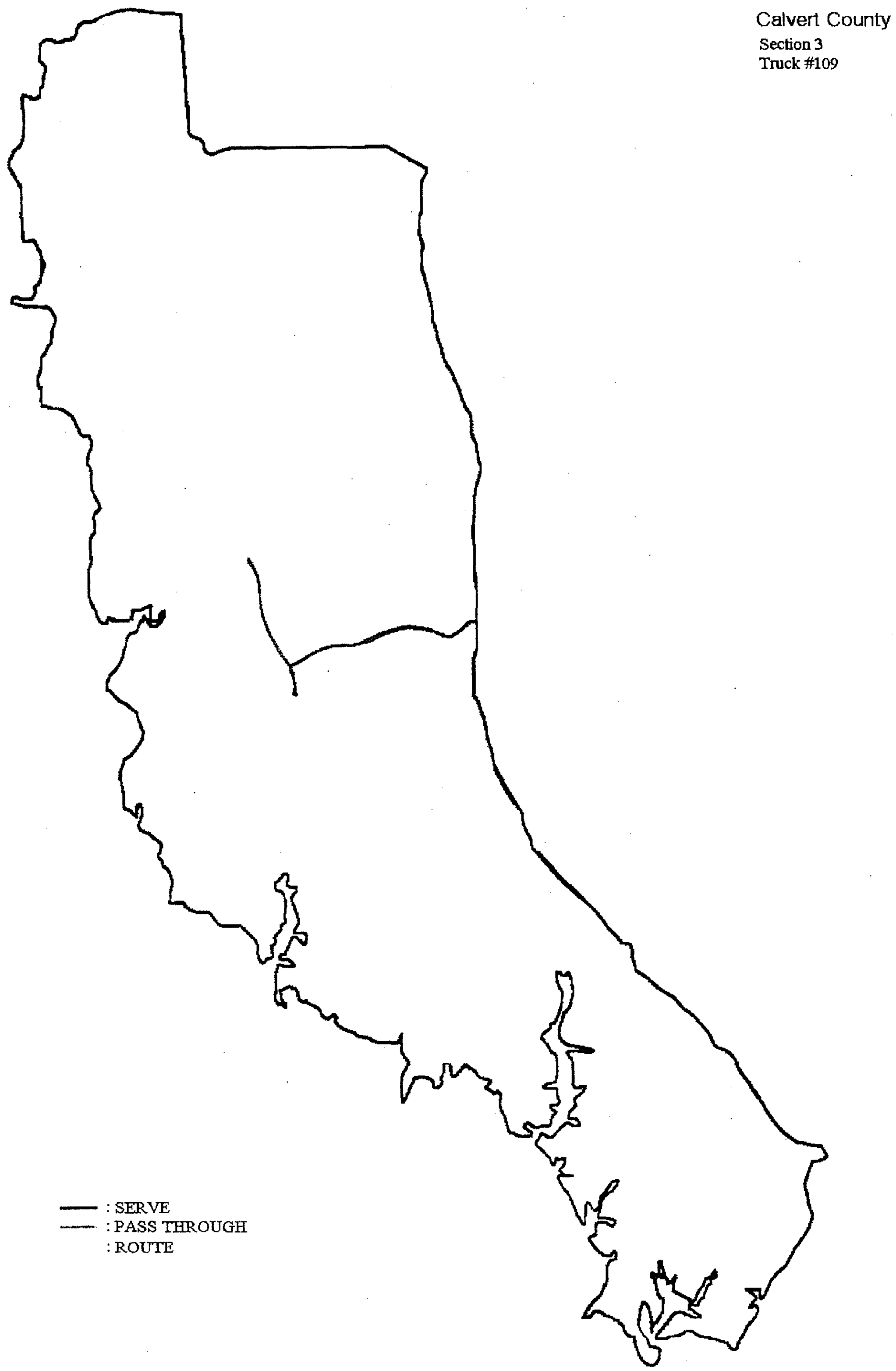


Figure 4.9 Calvert County Current Routes, Section III, Truck 109

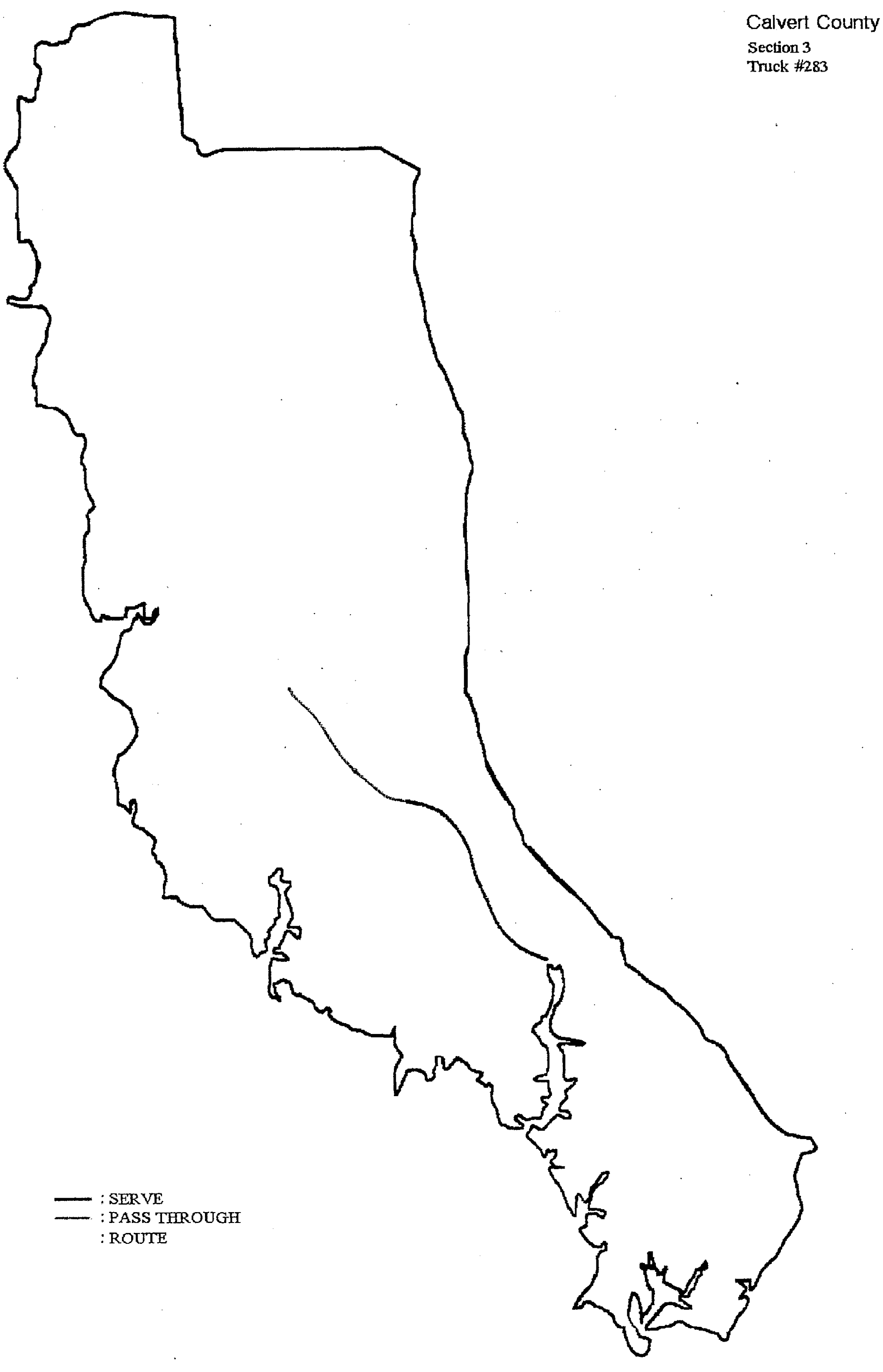


Figure 4.10 Calvert County Current Routes, Section III, Truck 283

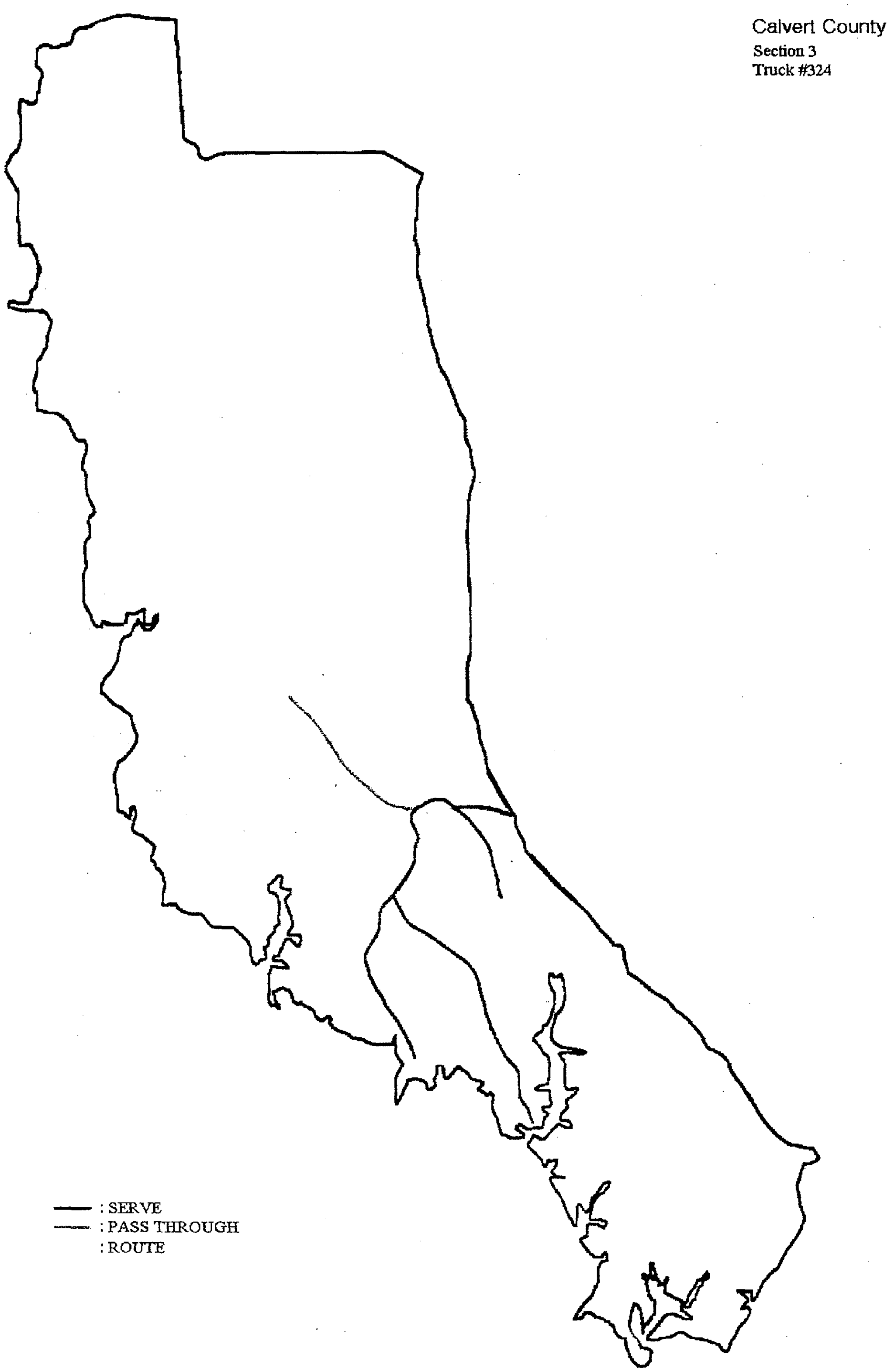


Figure 4.11 Calvert County Current Routes, Section III, Truck 324

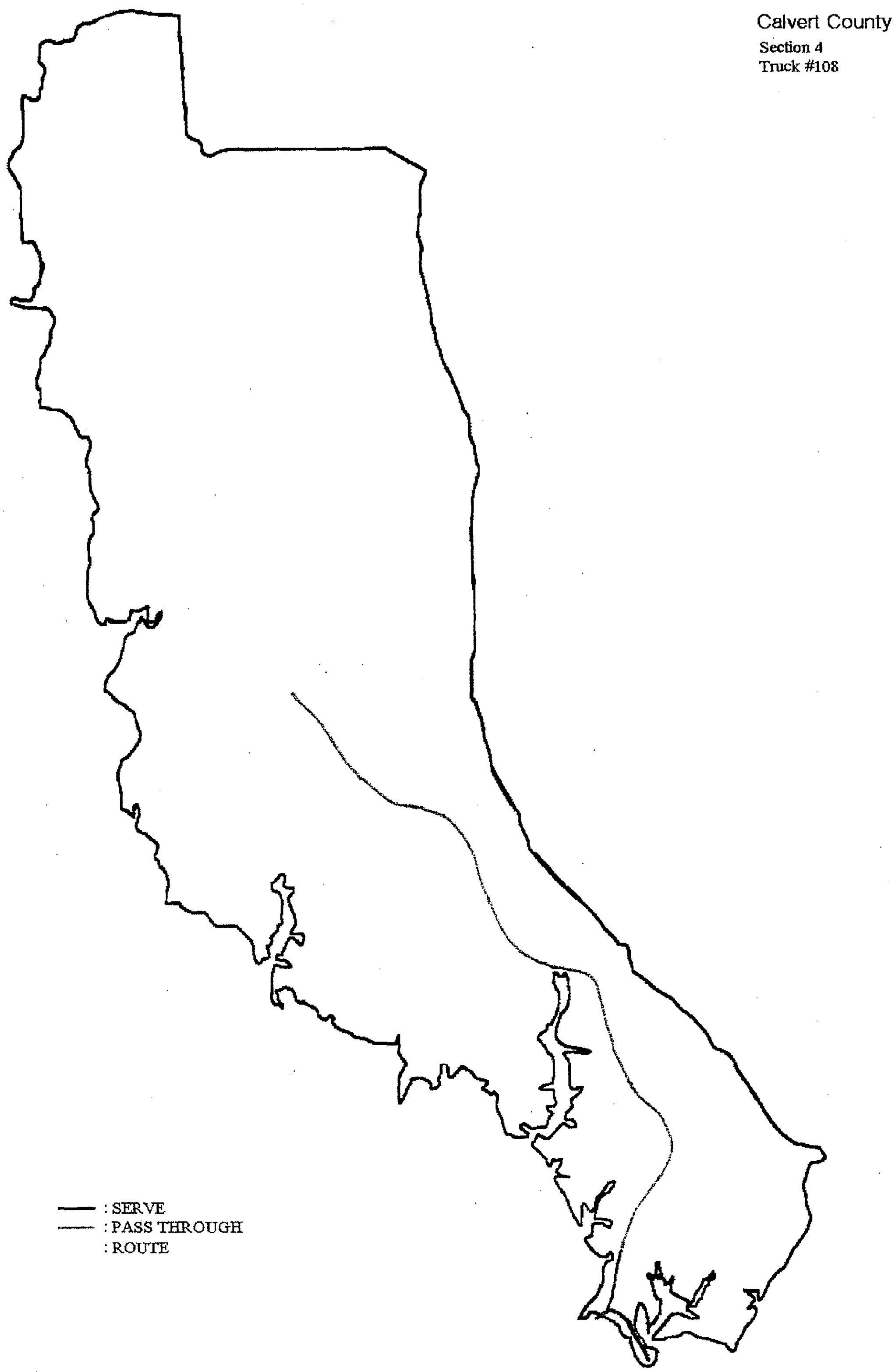


Figure 4.12 Calvert County Current Routes, Section IV, Truck 108

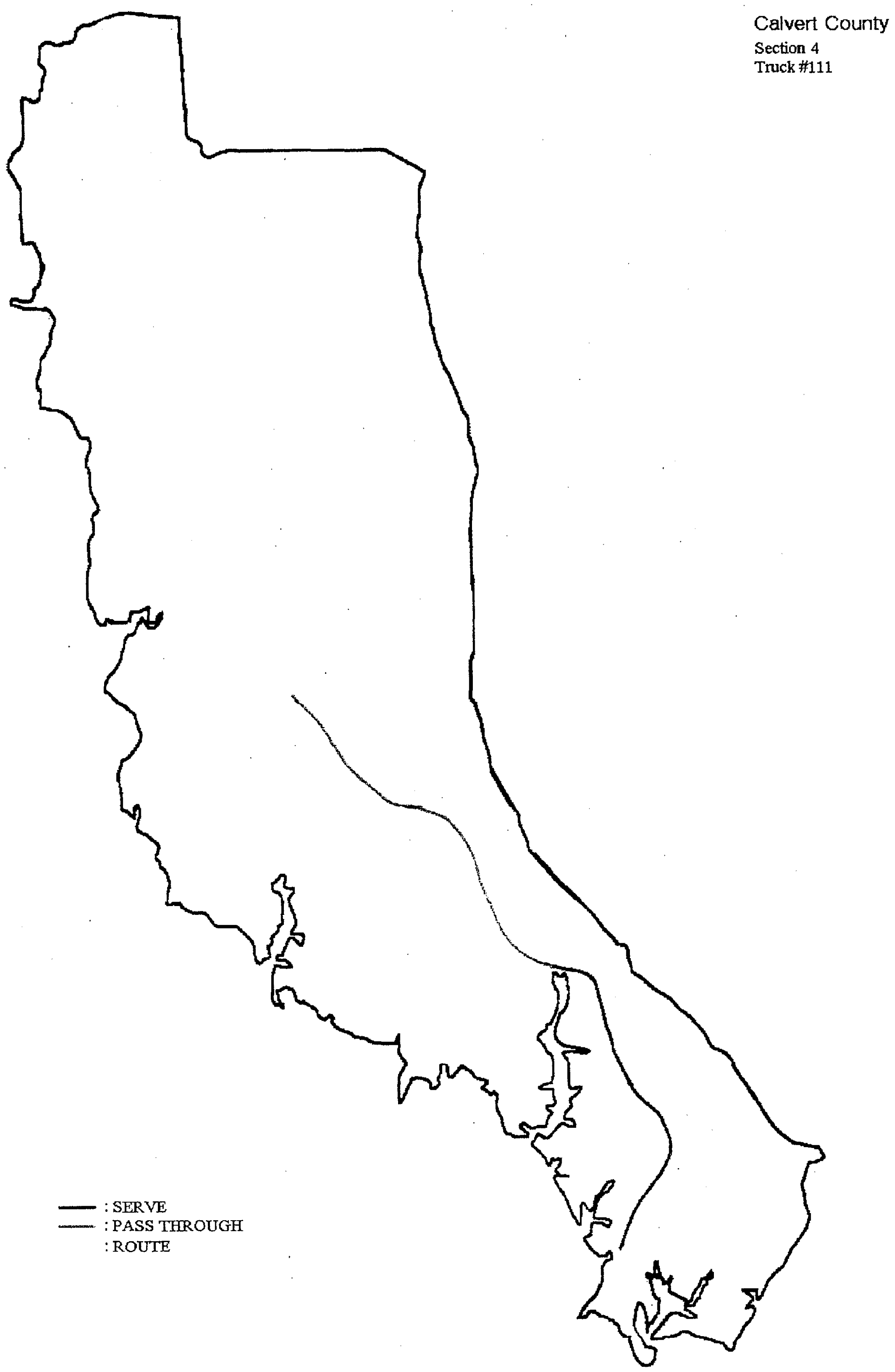


Figure 4.13 Calvert County Current Routes, Section IV, Truck 111

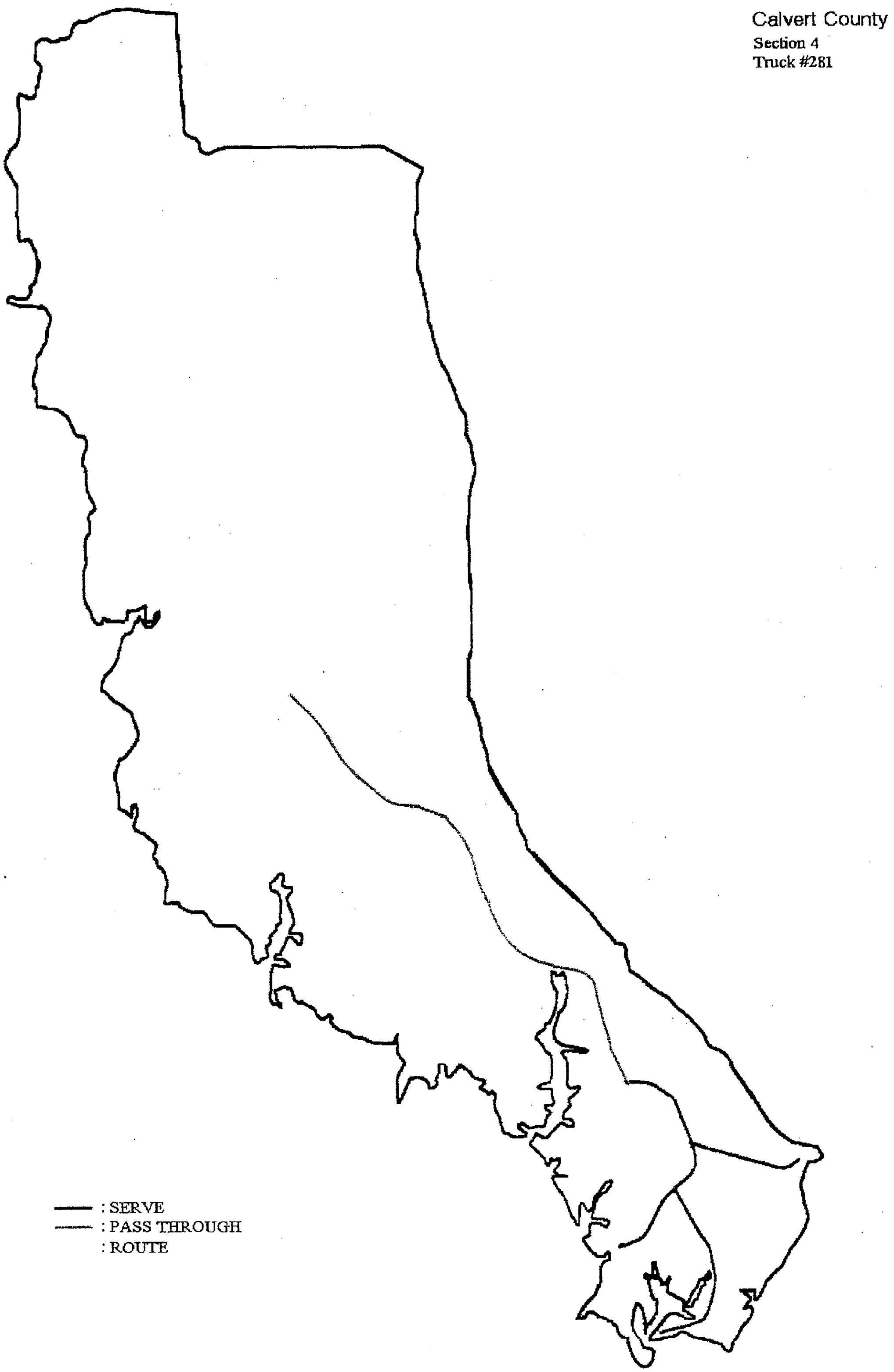


Figure 4.14 Calvert County Current Routes, Section IV, Truck 281

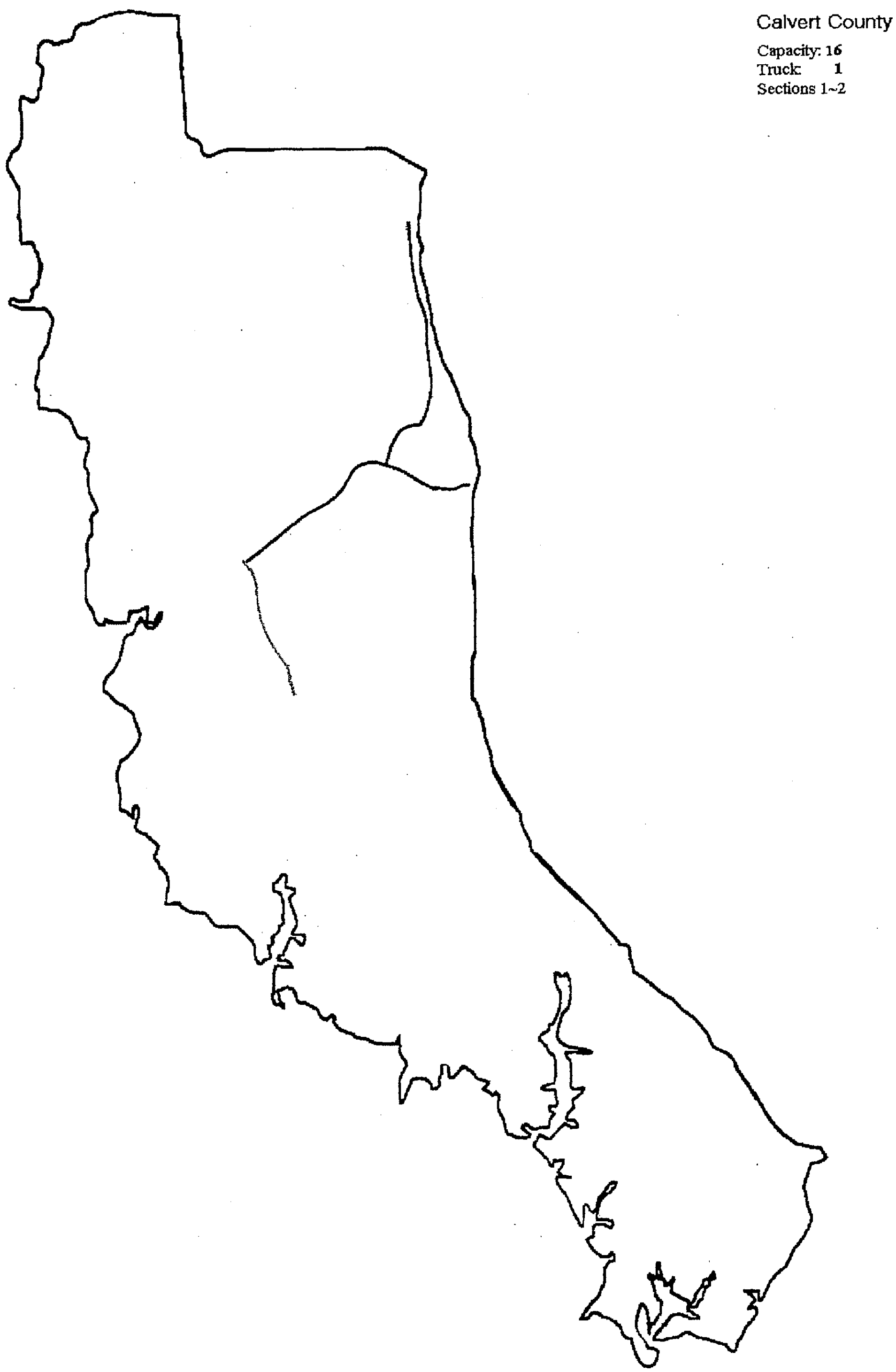


Figure 4.15 Sections I & II, Service Route Length 16 miles

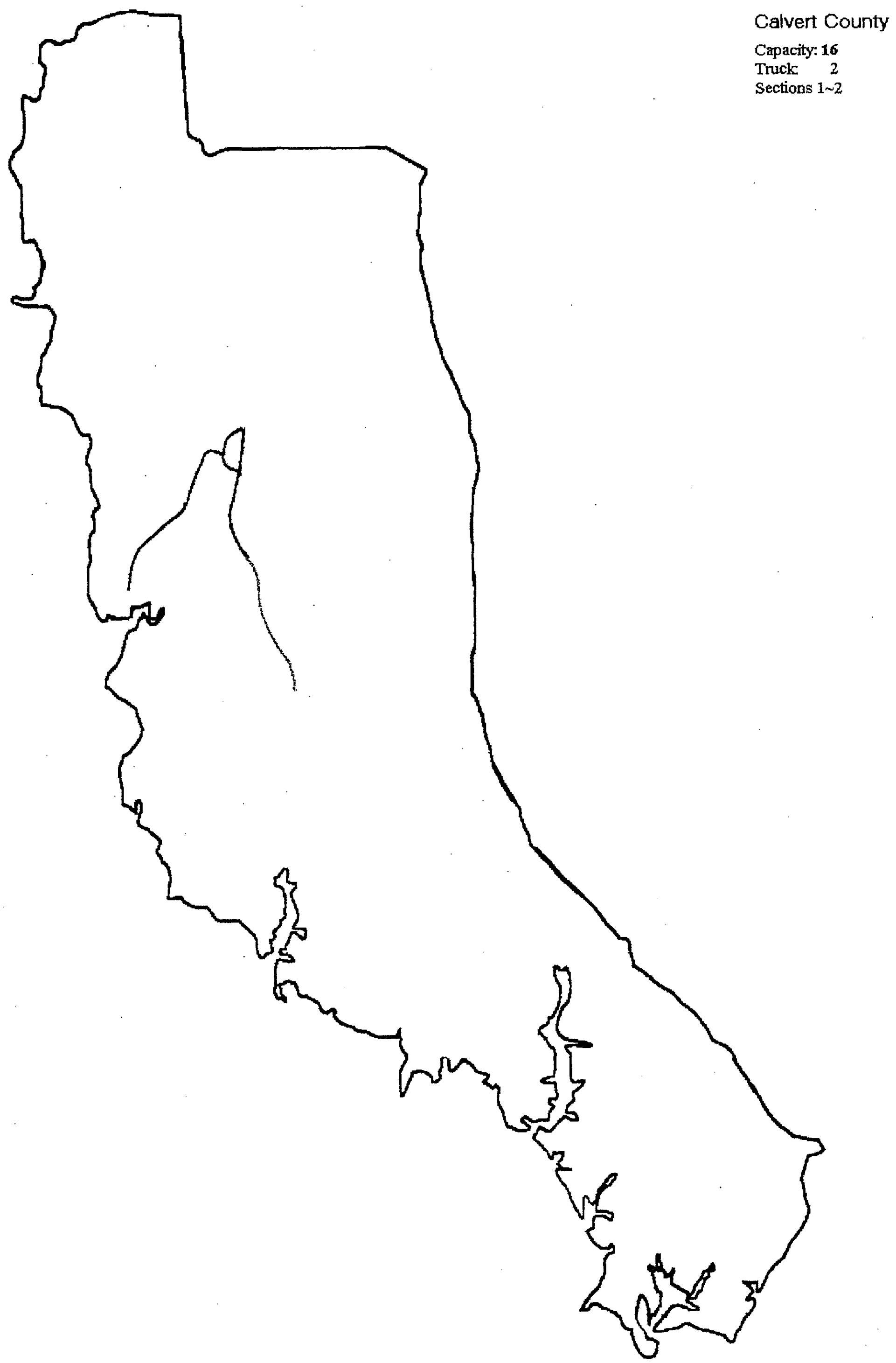


Figure 4.16 Sections I & II, Service Route Length 16 miles

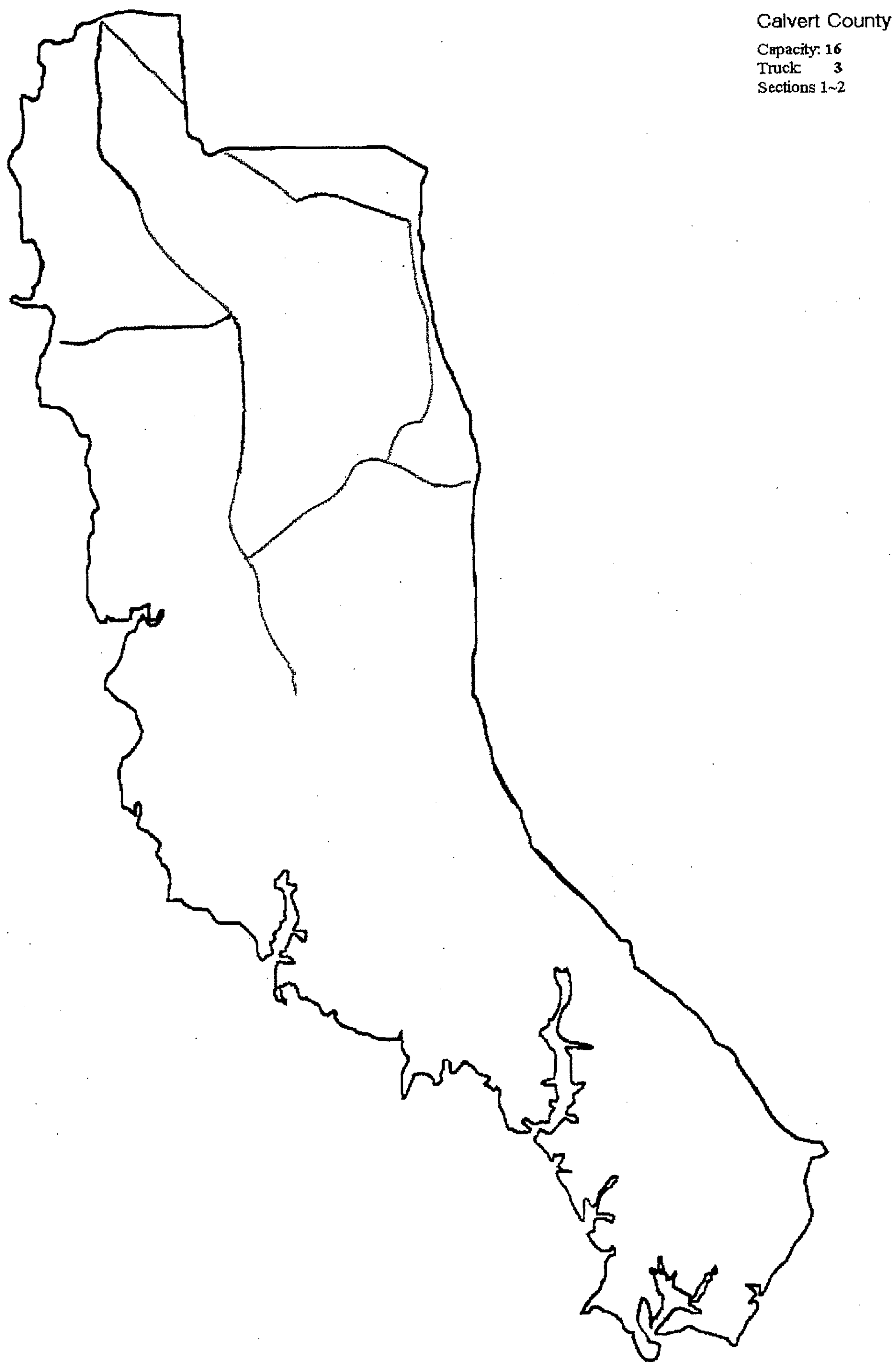


Figure 4.17 Sections I & II, Service Route Length 16 miles

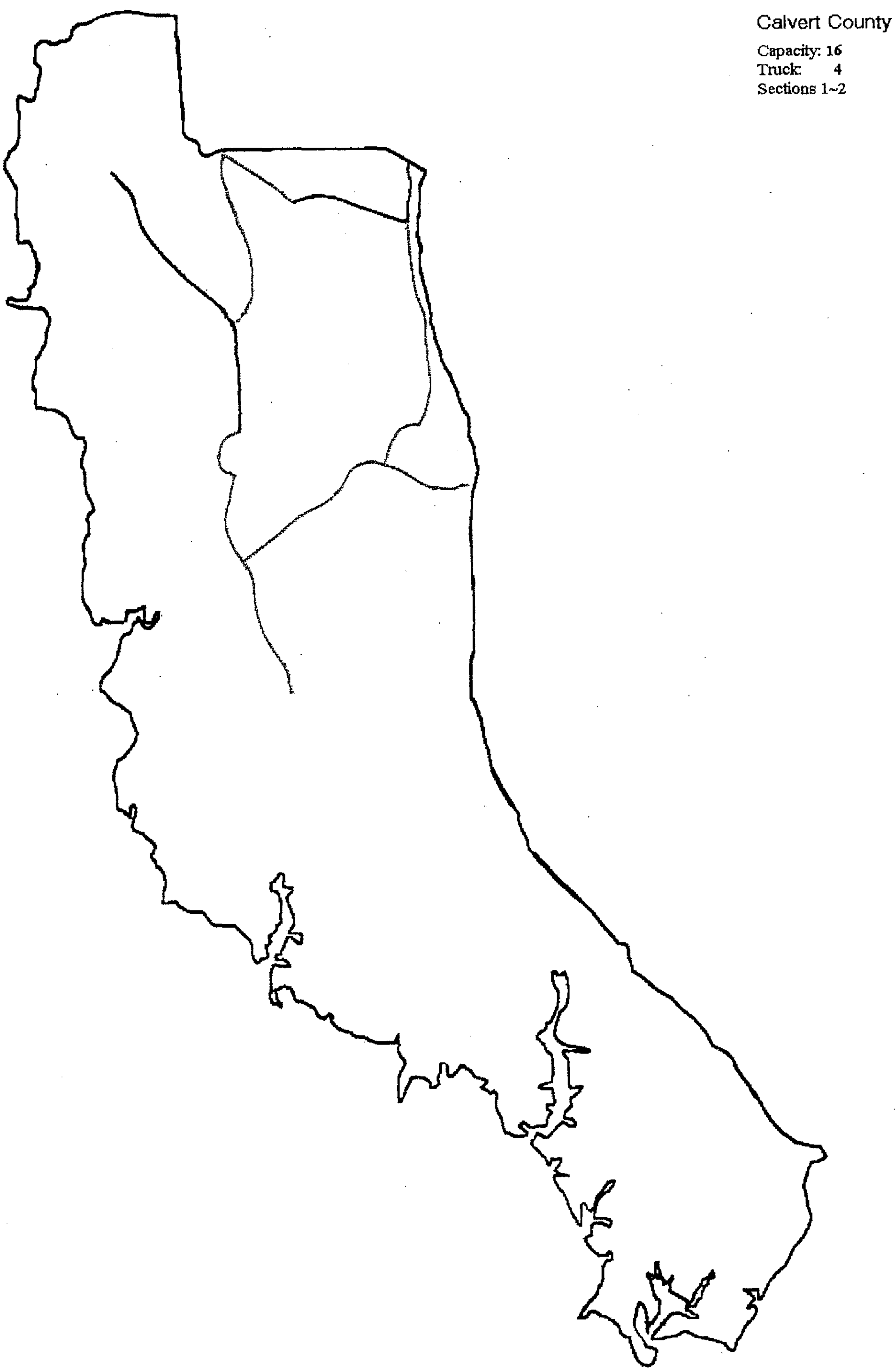


Figure 4.18 Sections I & II, Service Route Length 16 miles

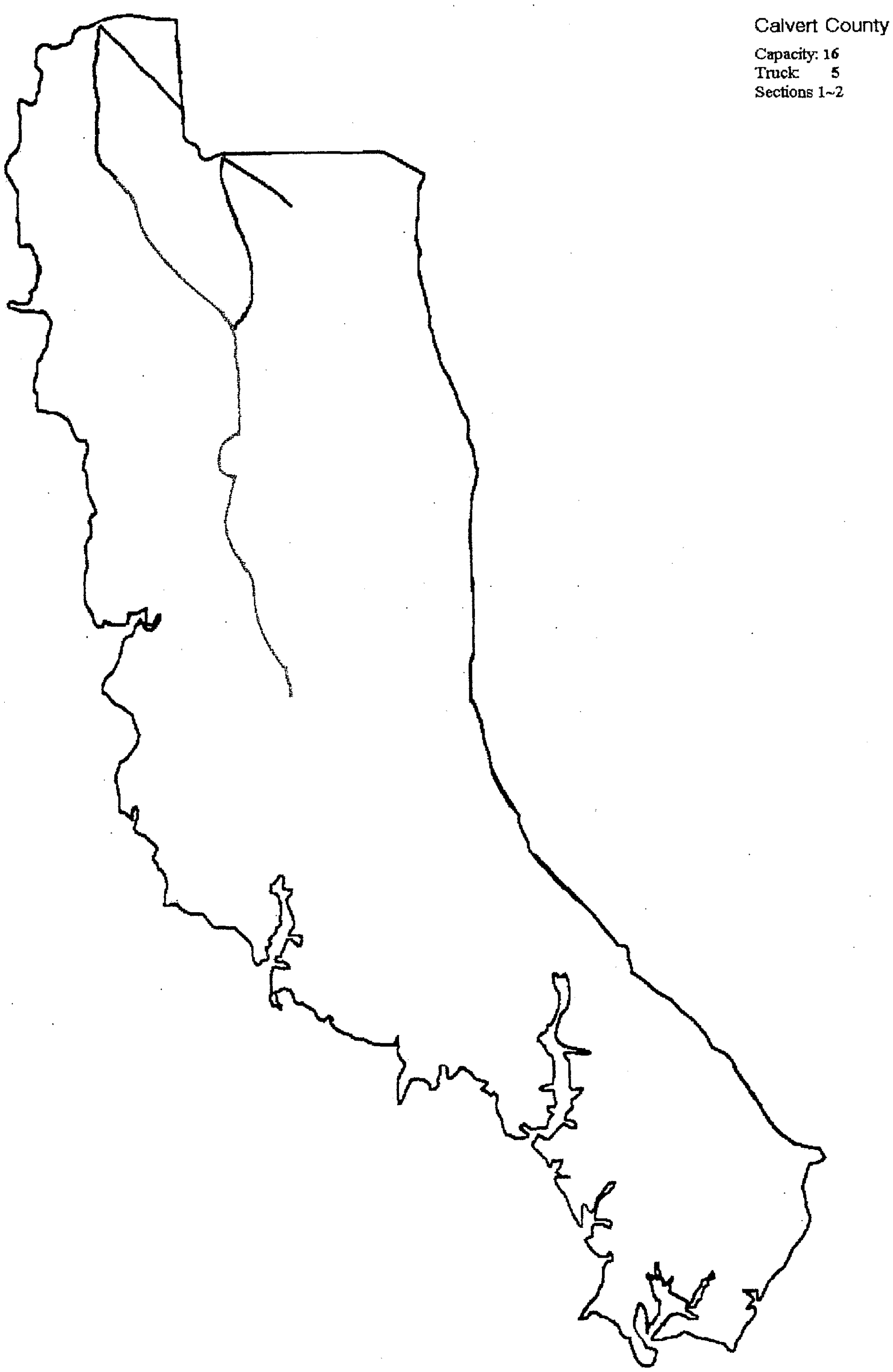


Figure 4.19 Sections I & II, Service Route Length 16 miles

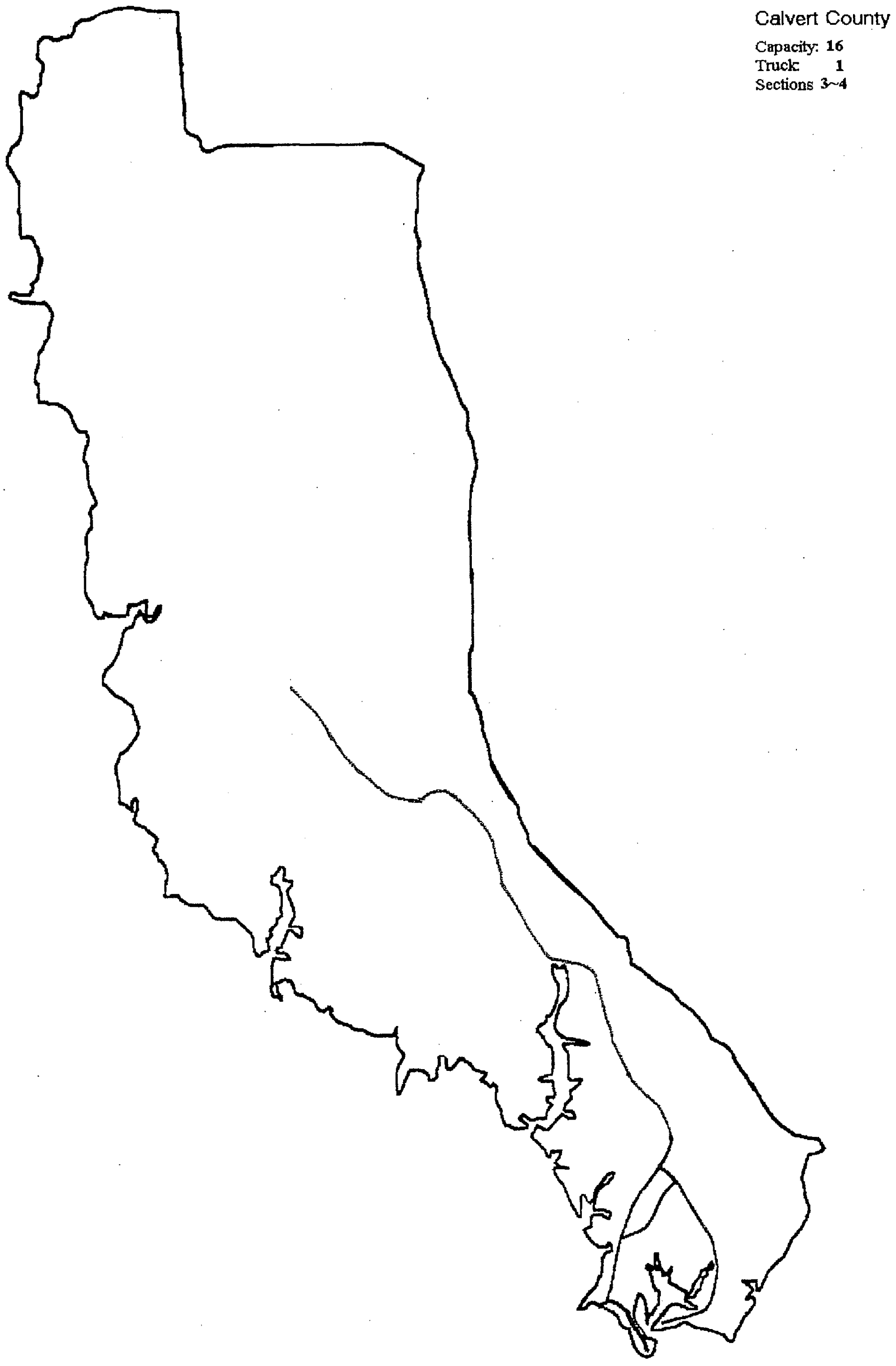


Figure 4.20 Sections III & IV, Service Route Length 16 miles

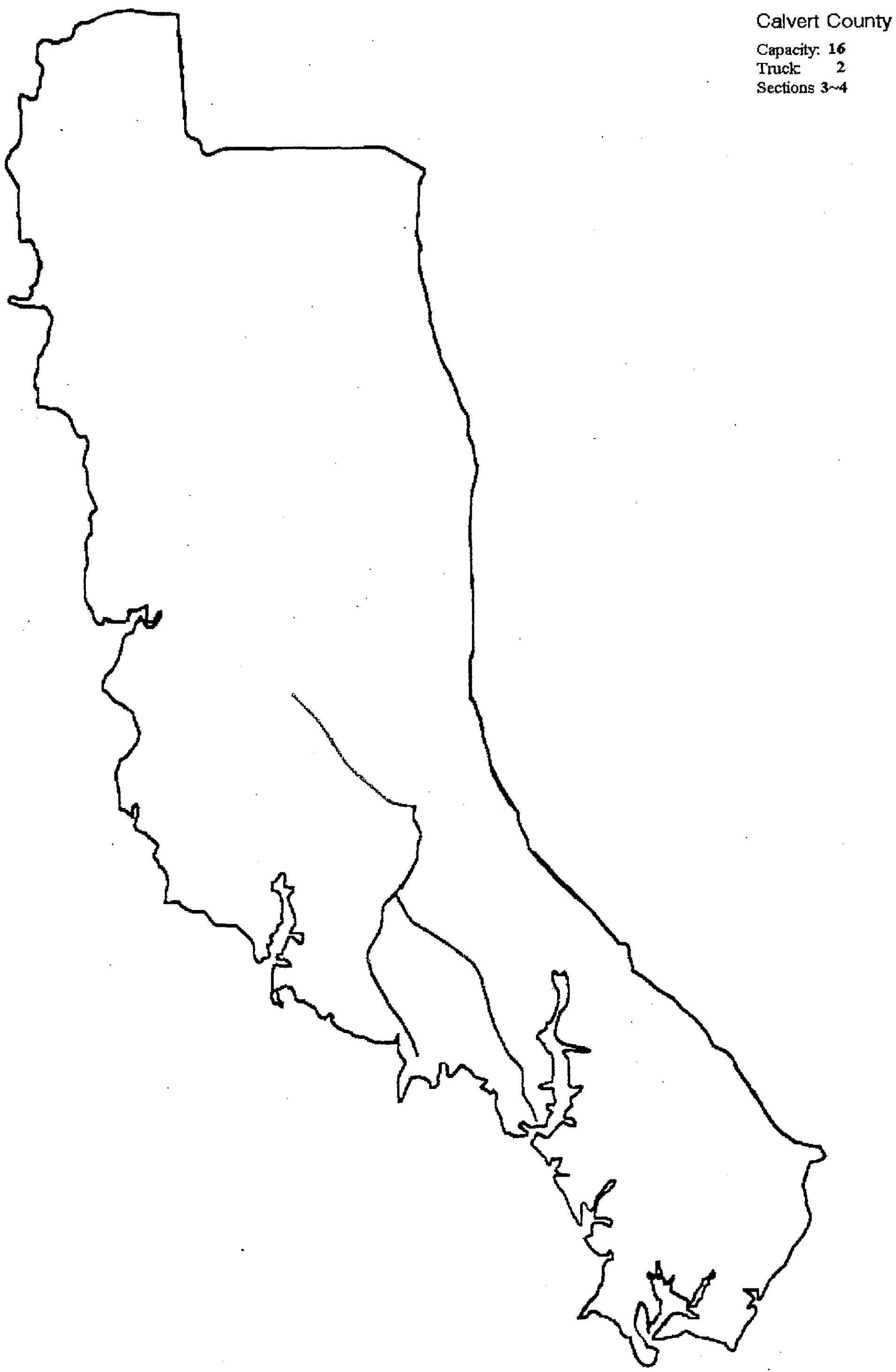


Figure 4.21 Sections III & IV, Service Route Length 16 miles

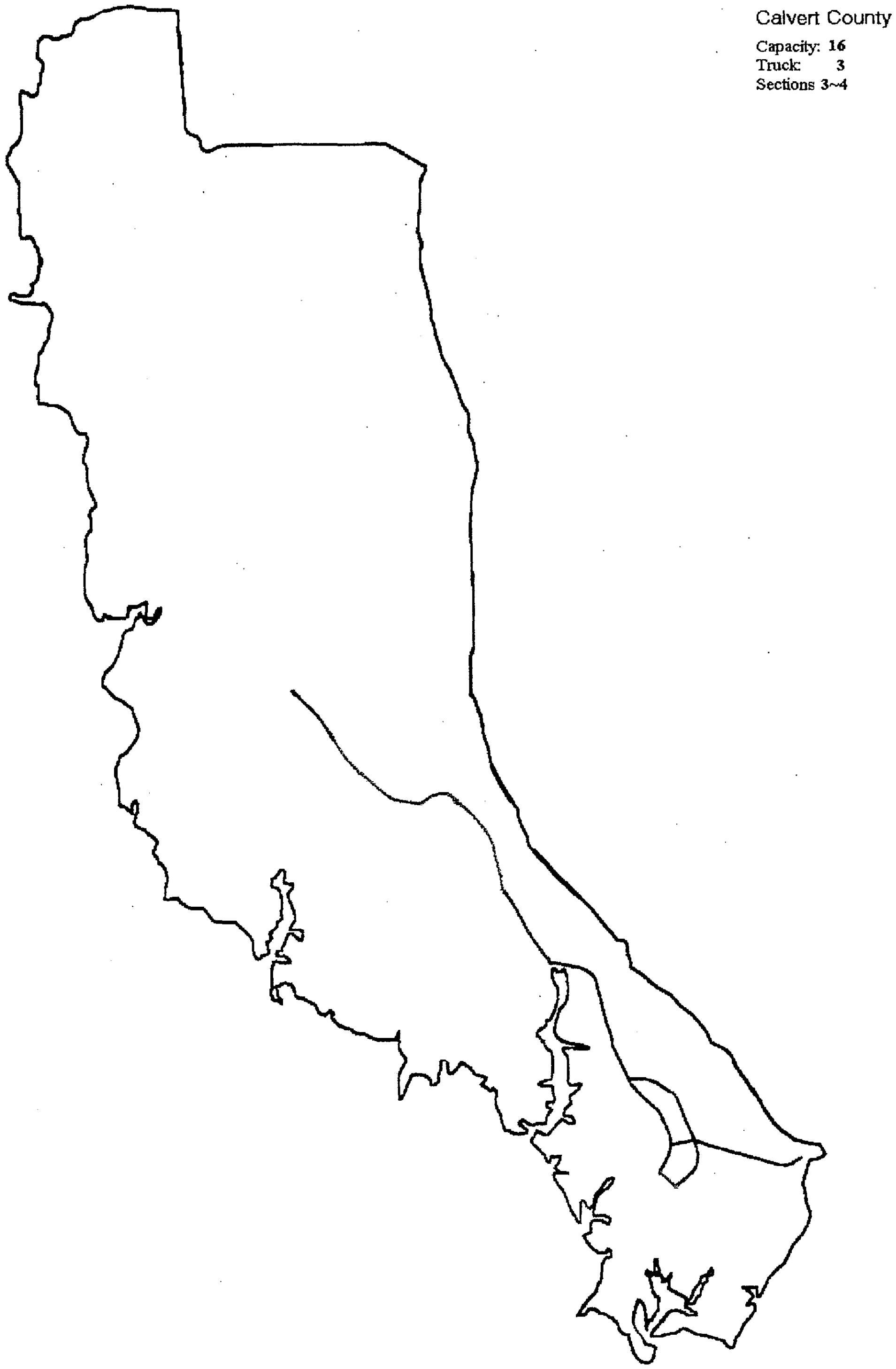


Figure 4.22 Sections III & IV, Service Route Length 16 miles

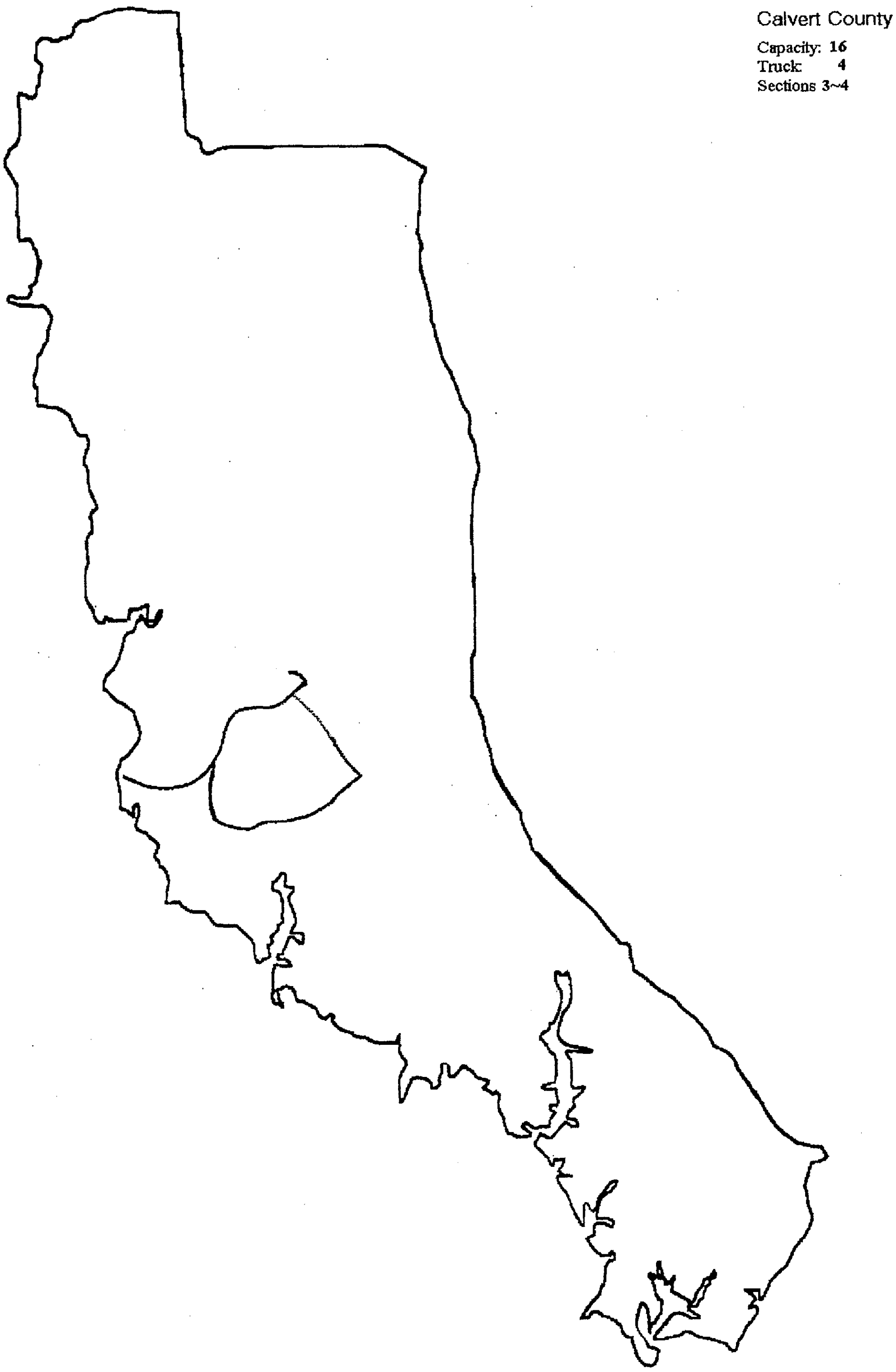


Figure 4.23 Sections III & IV, Service Route Length 16 miles

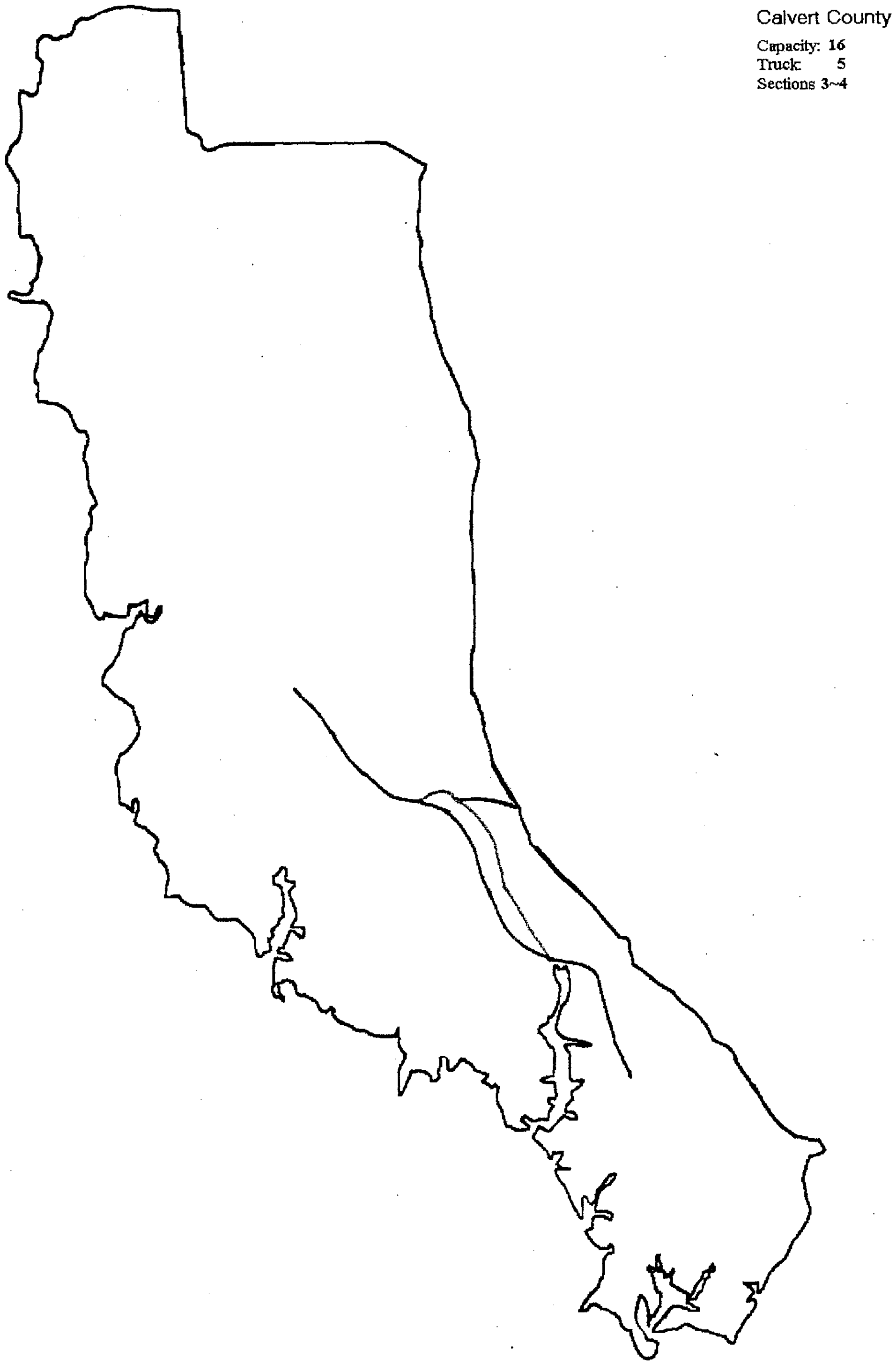


Figure 4.24 Sections III & IV, Service Route Length 16 miles

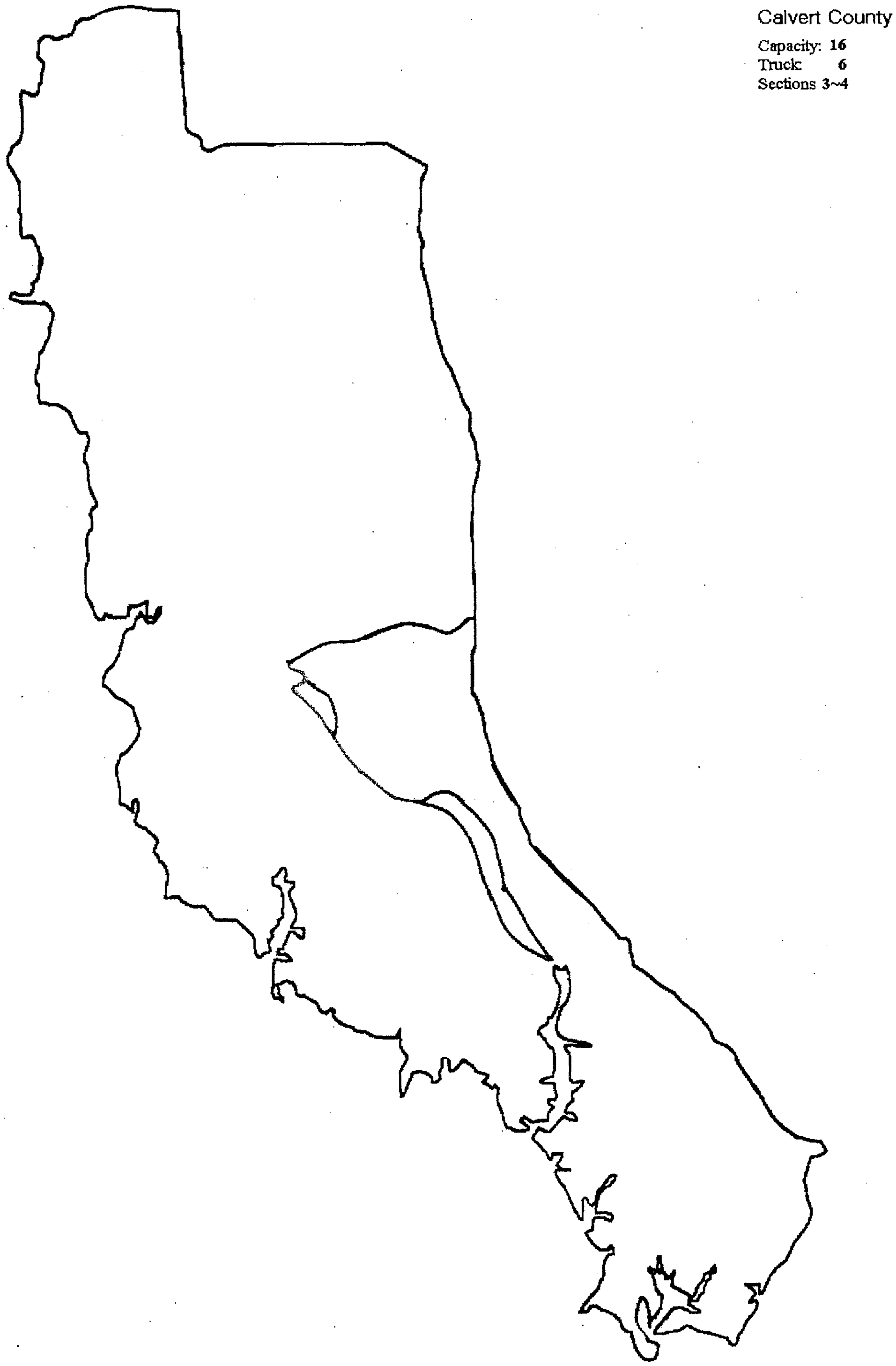


Figure 4.25 Sections III & IV, Service Route Length 16 miles

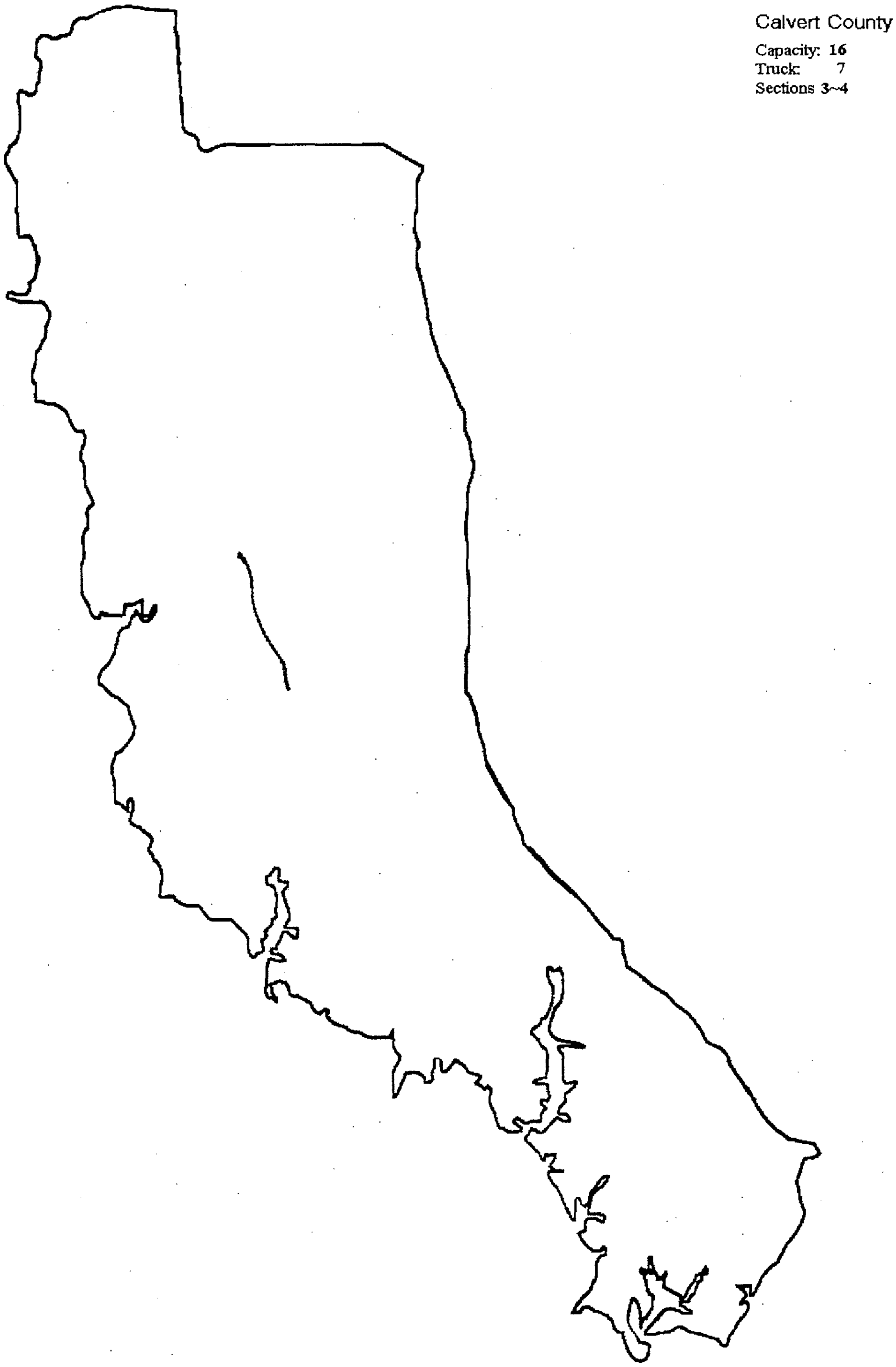


Figure 4.26 Sections IIII& IV, Service Route Length 16 miles

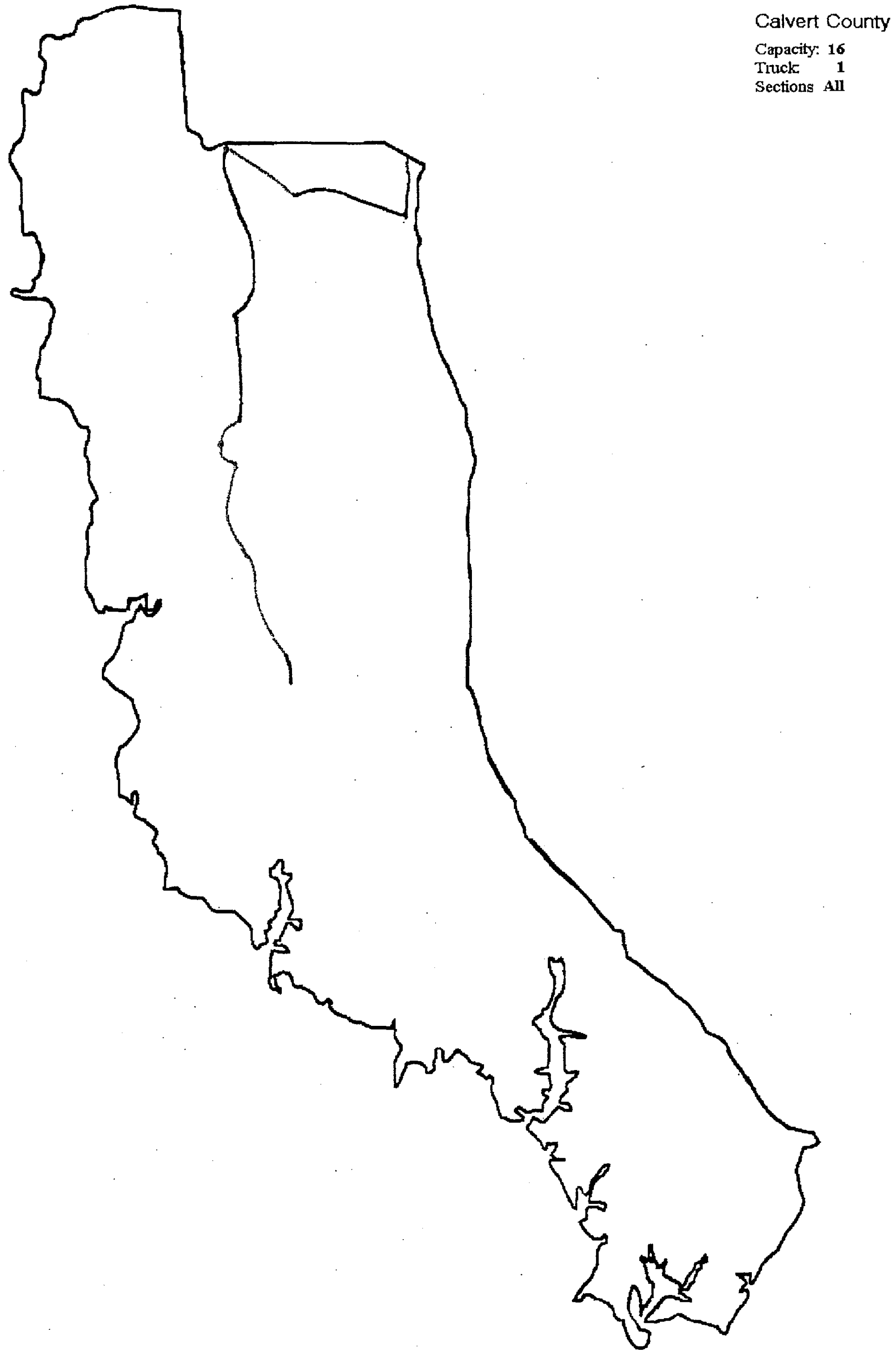


Figure 4.27 All Sections, Service Route Length 16 miles

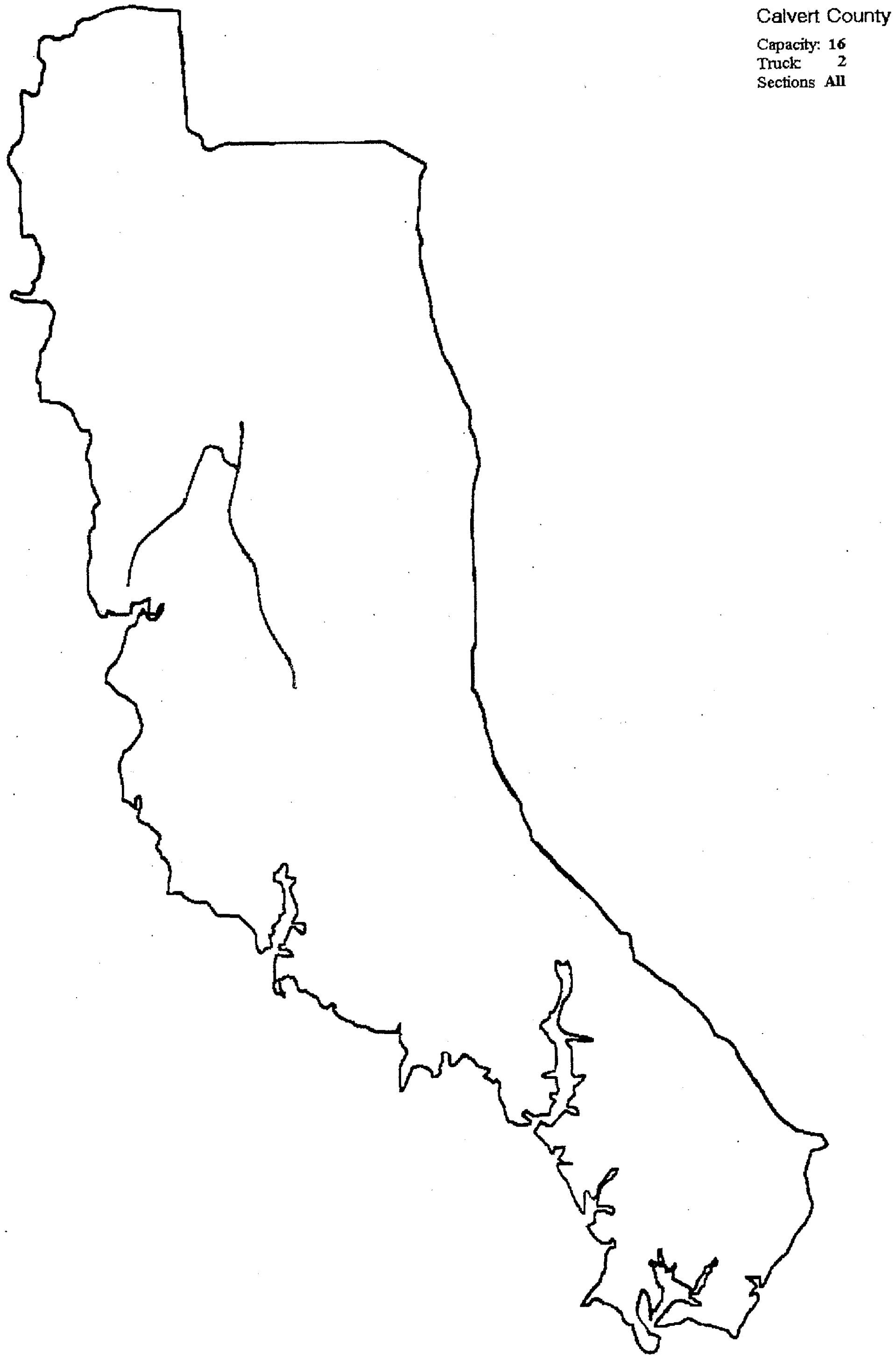


Figure 4.28 All Sections, Service Route Length 16 miles

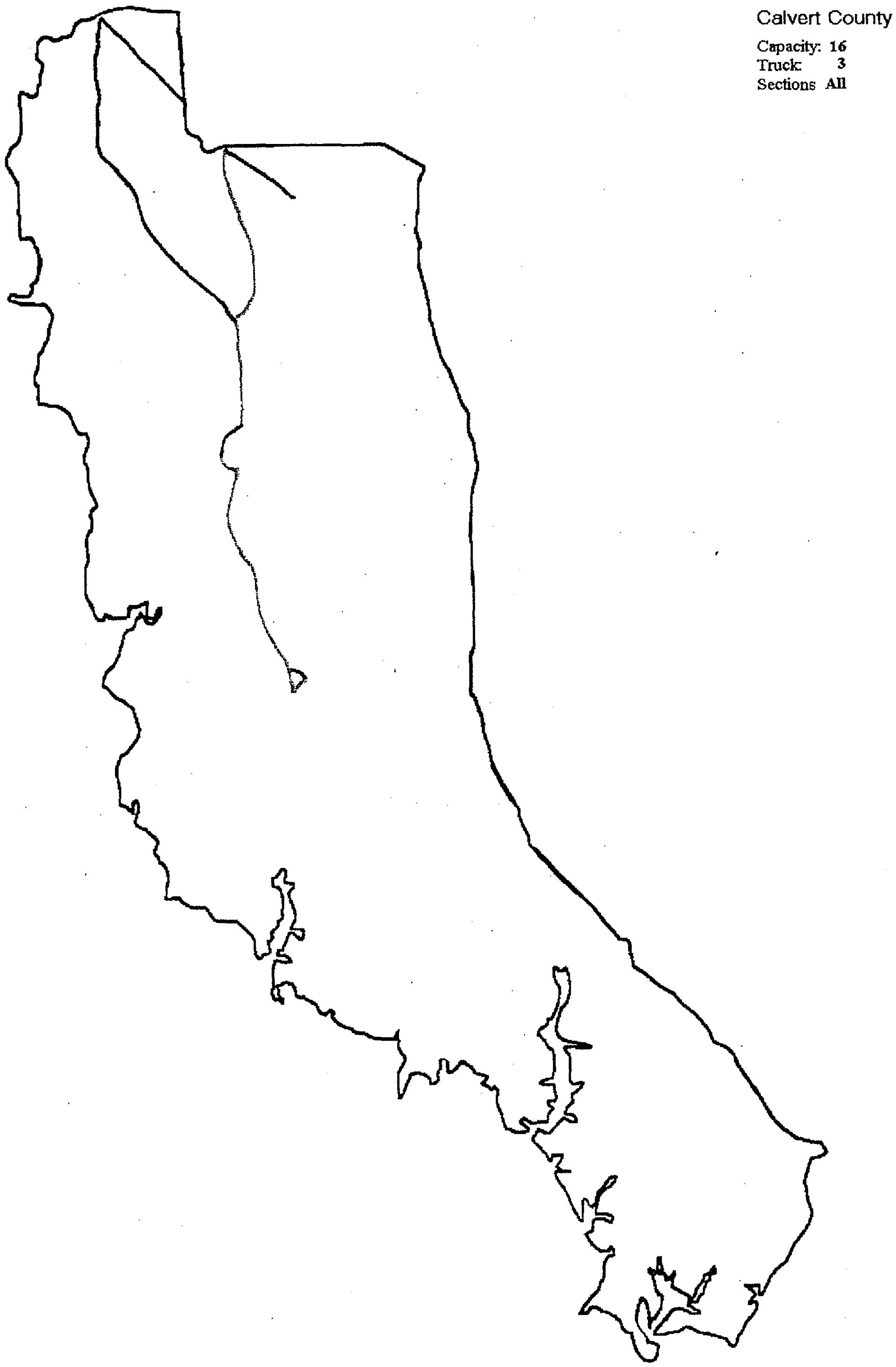


Figure 4.29 All Sections, Service Route Length 16 miles

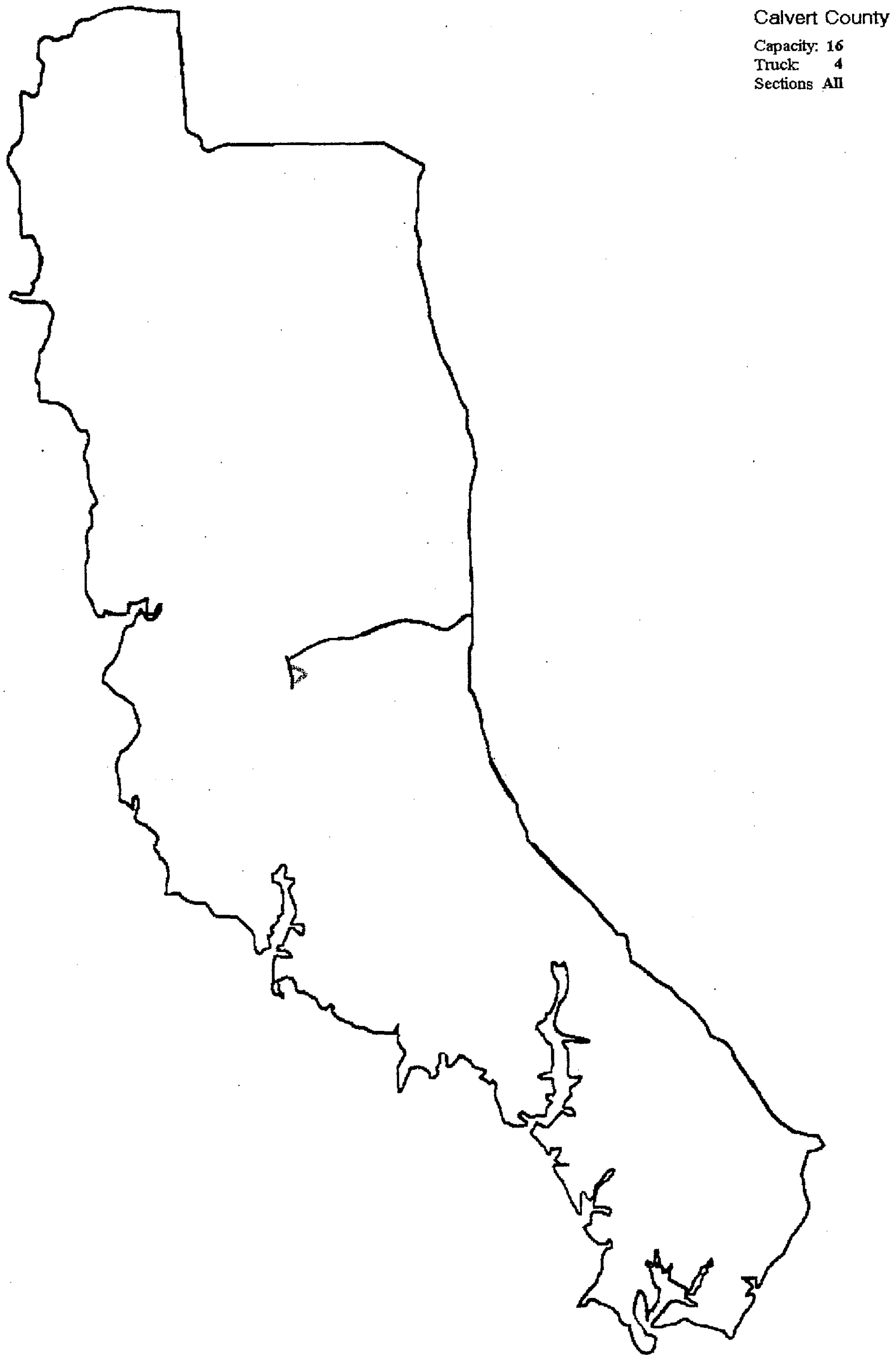


Figure 4.30 All Sections, Service Route Length 16 miles

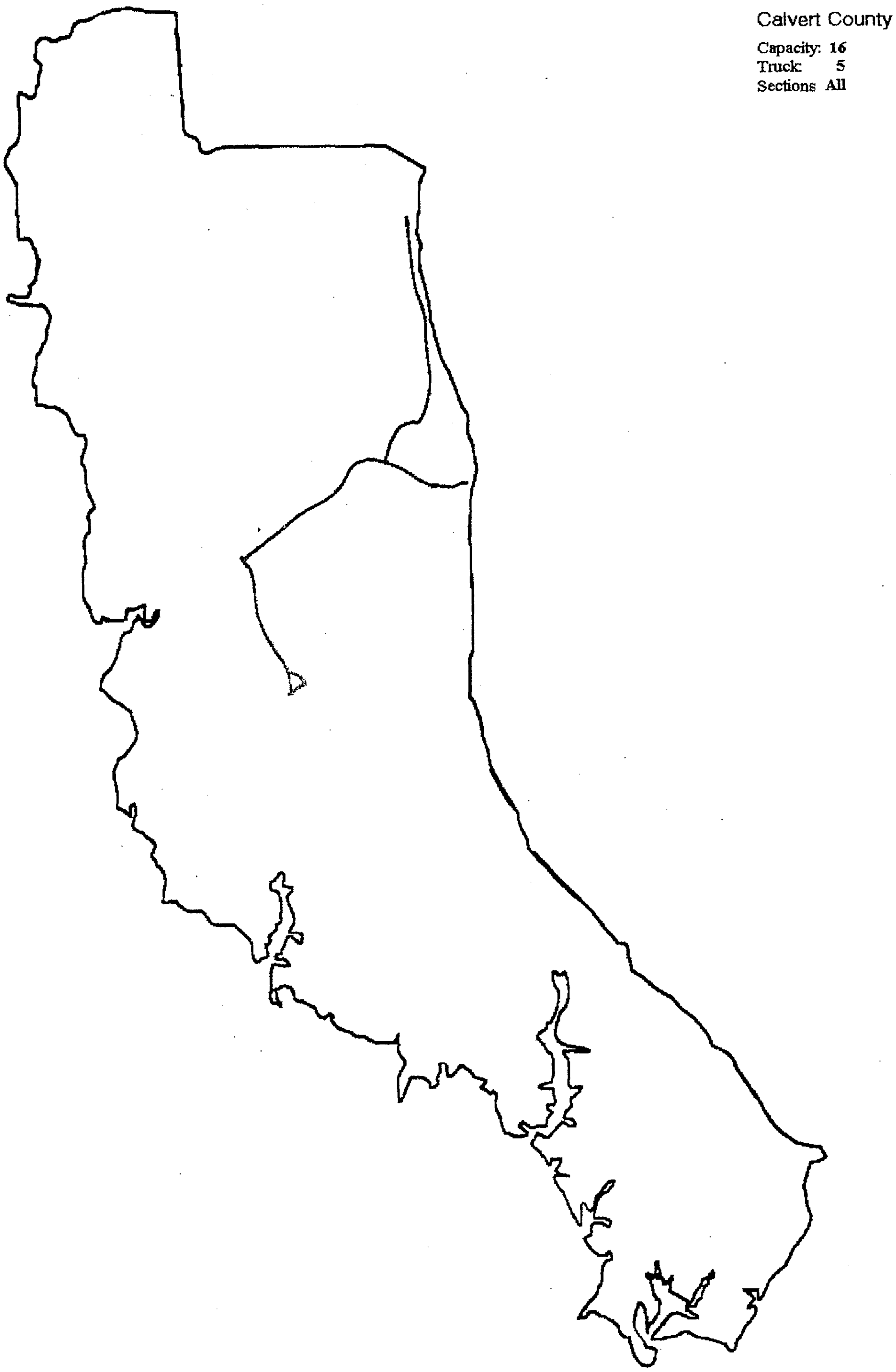


Figure 4.31 All Sections, Service Route Length 16 miles

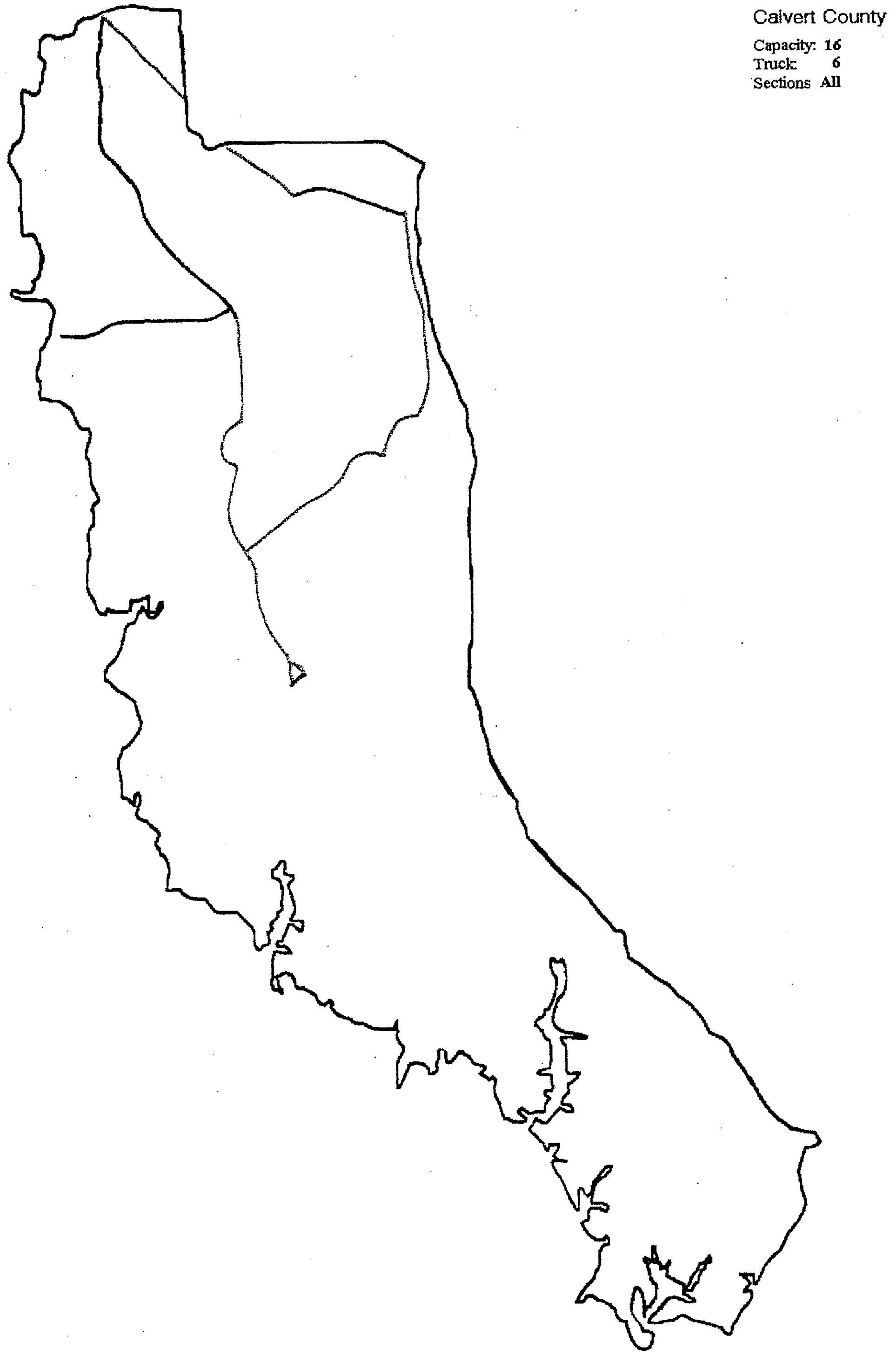


Figure 4.32 All Sections, Service Route Length 16 miles

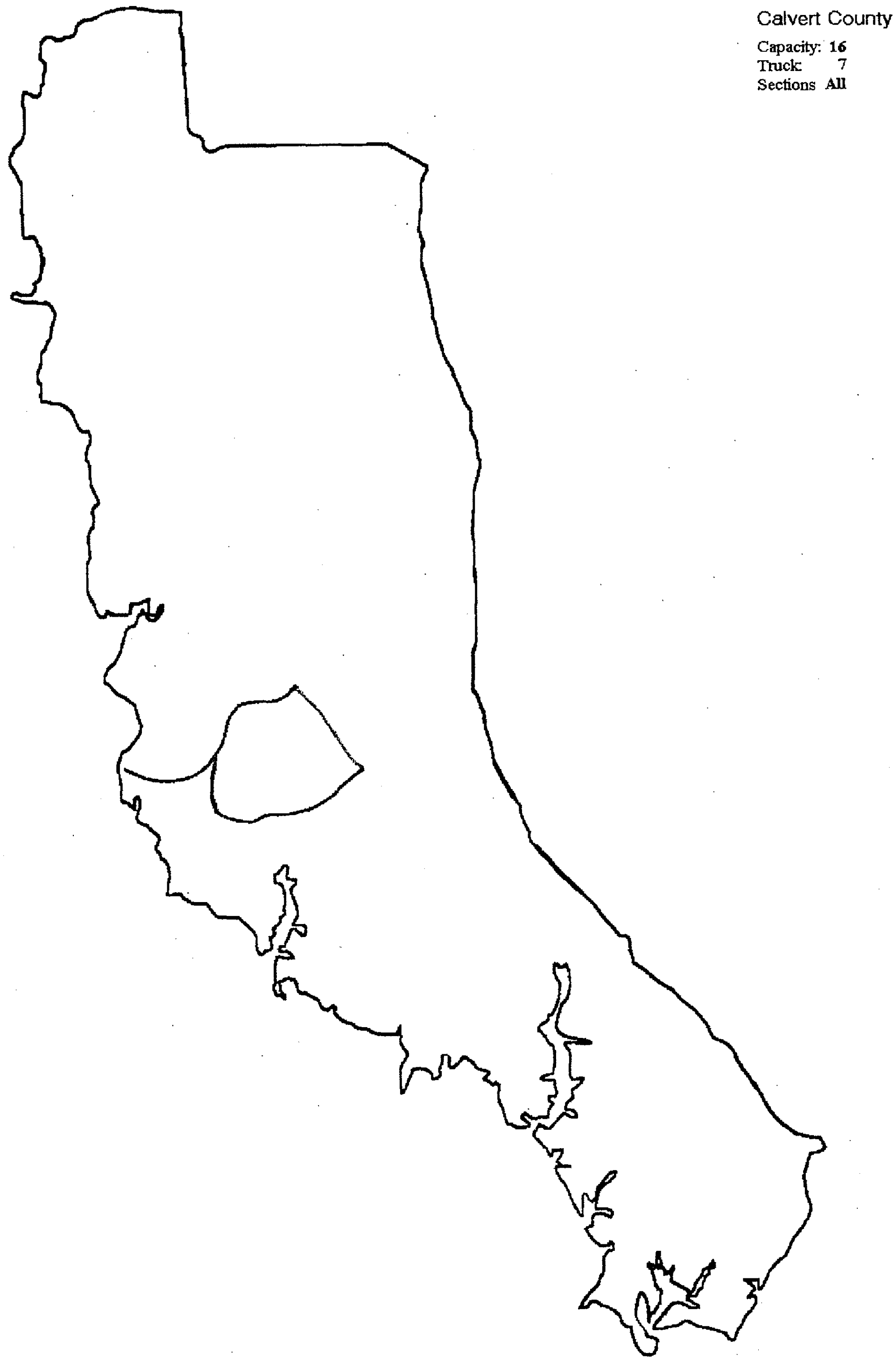


Figure 4.33 All Sections, Service Route Length 16 miles

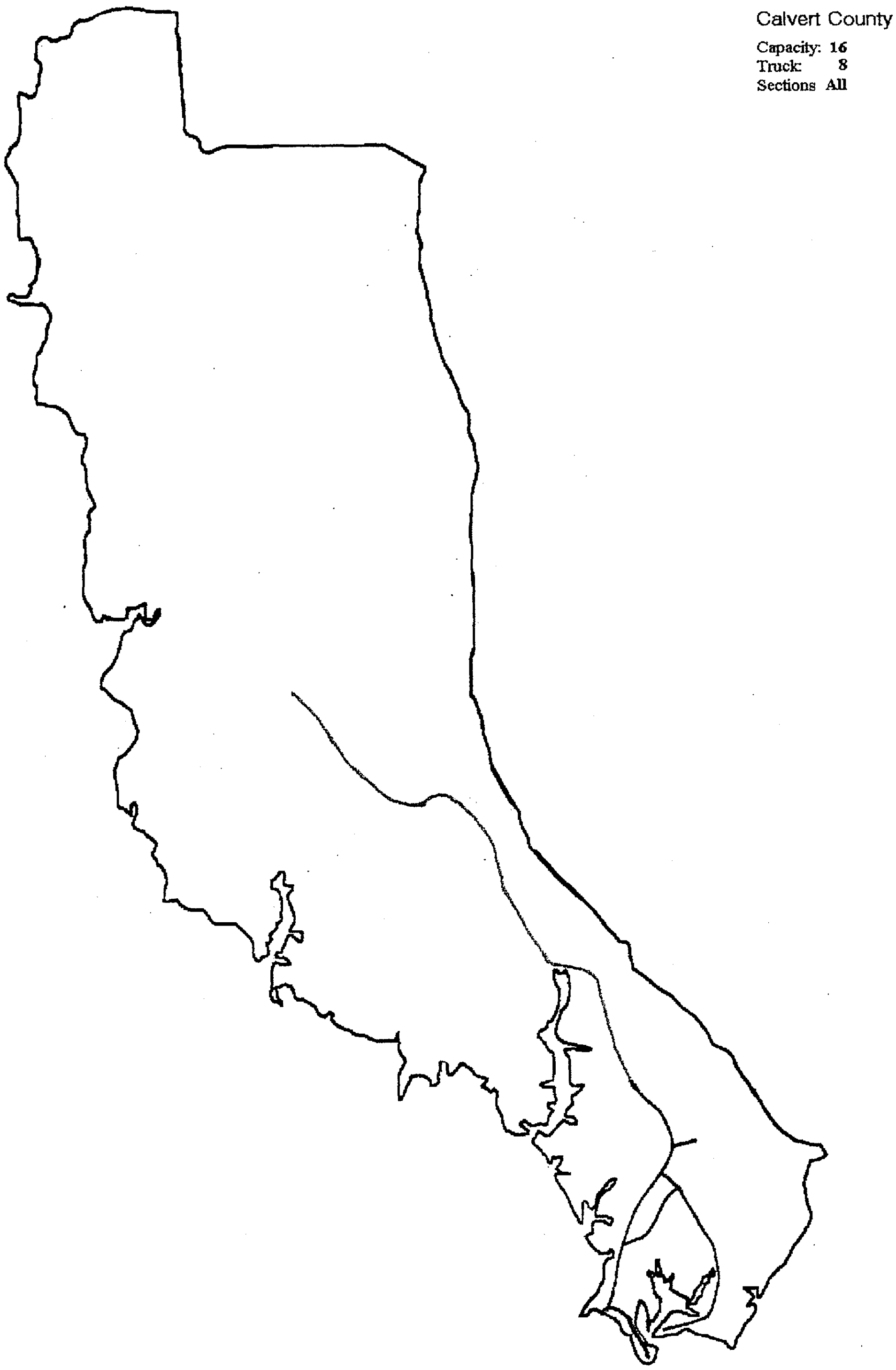


Figure 4.34 All Sections, Service Route Length 16 miles

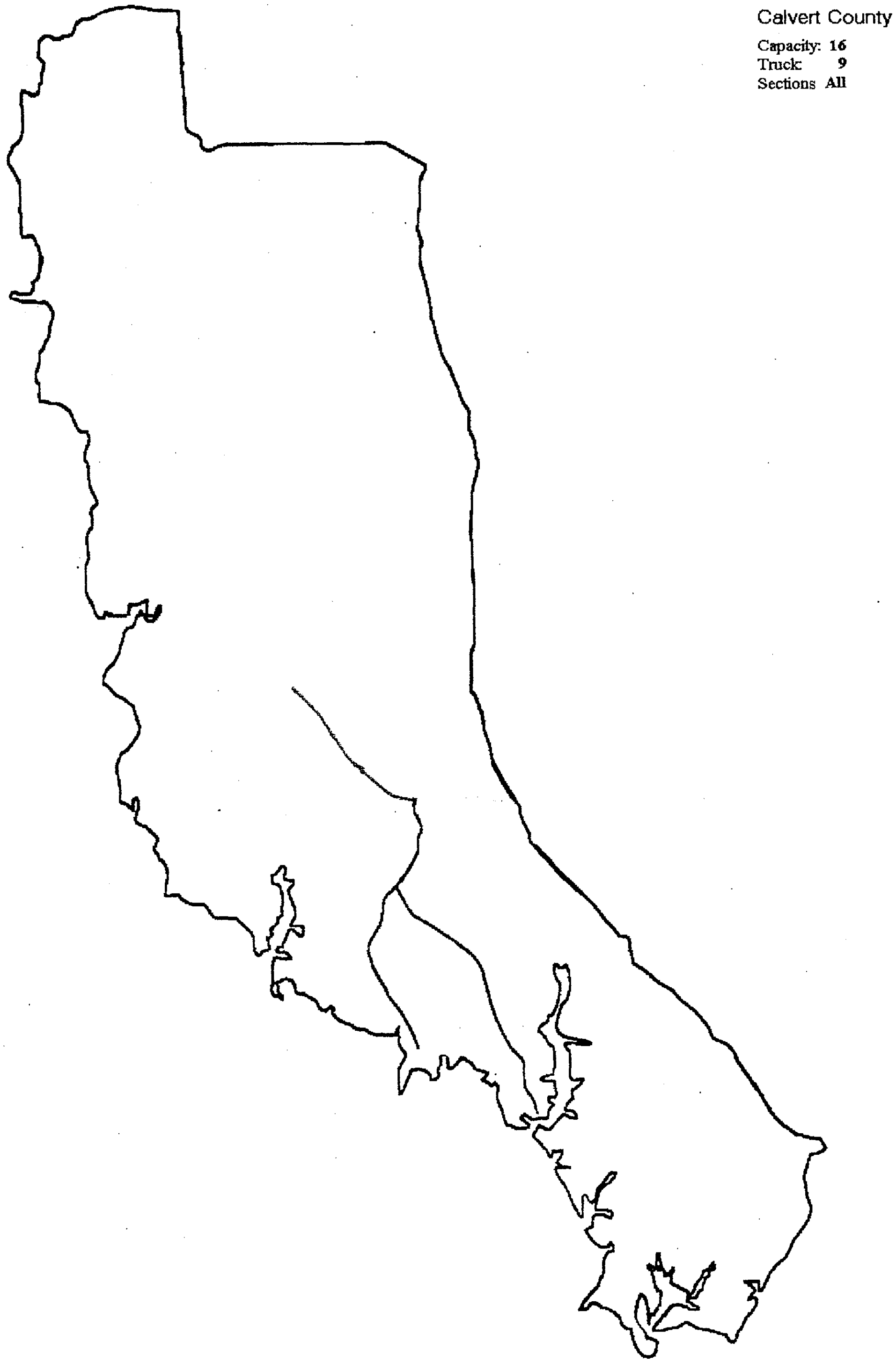


Figure 4.35 All Sections, Service Route Length 16 miles

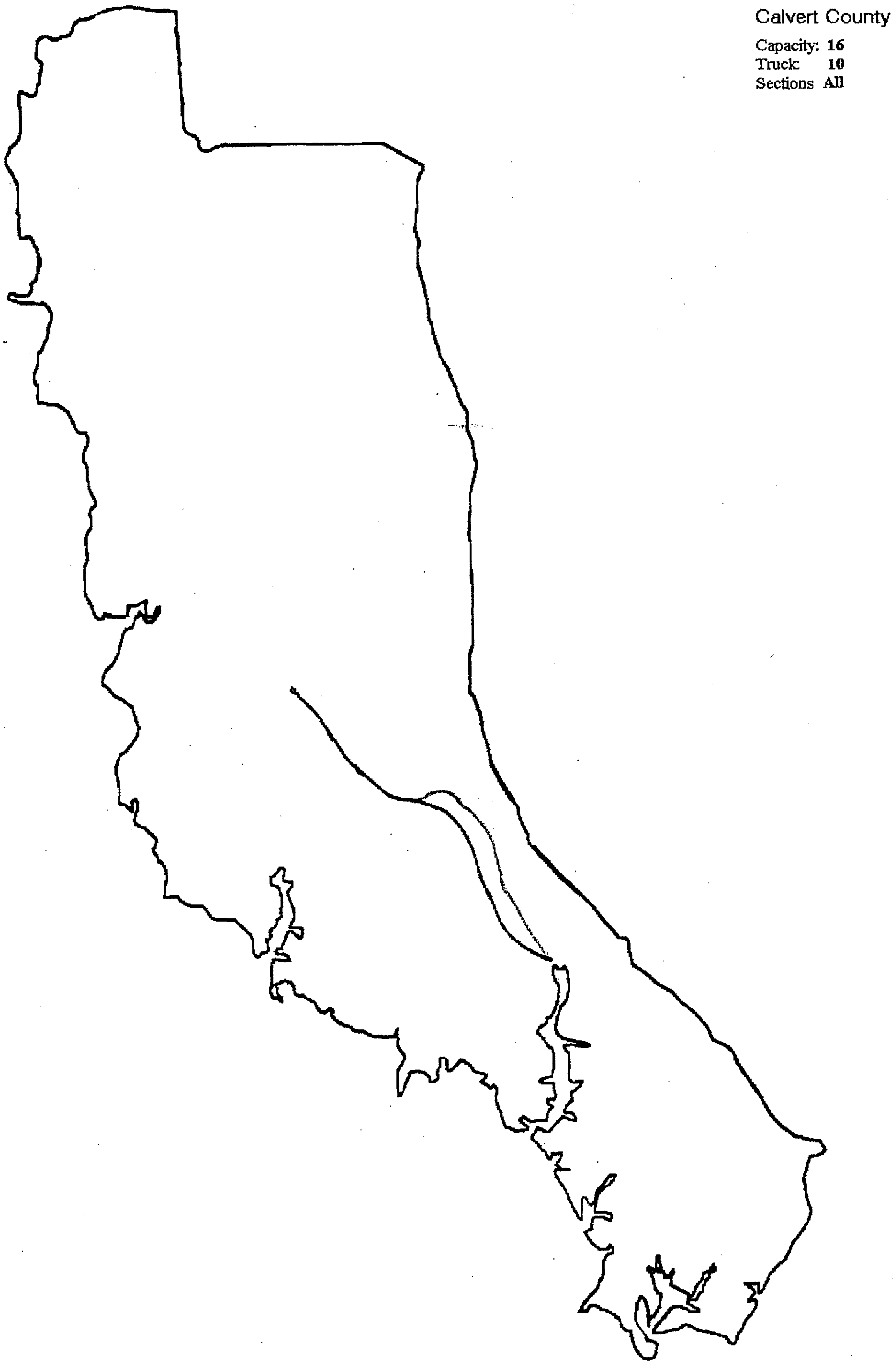


Figure 4.36 All Sections, Service Route Length 16 miles

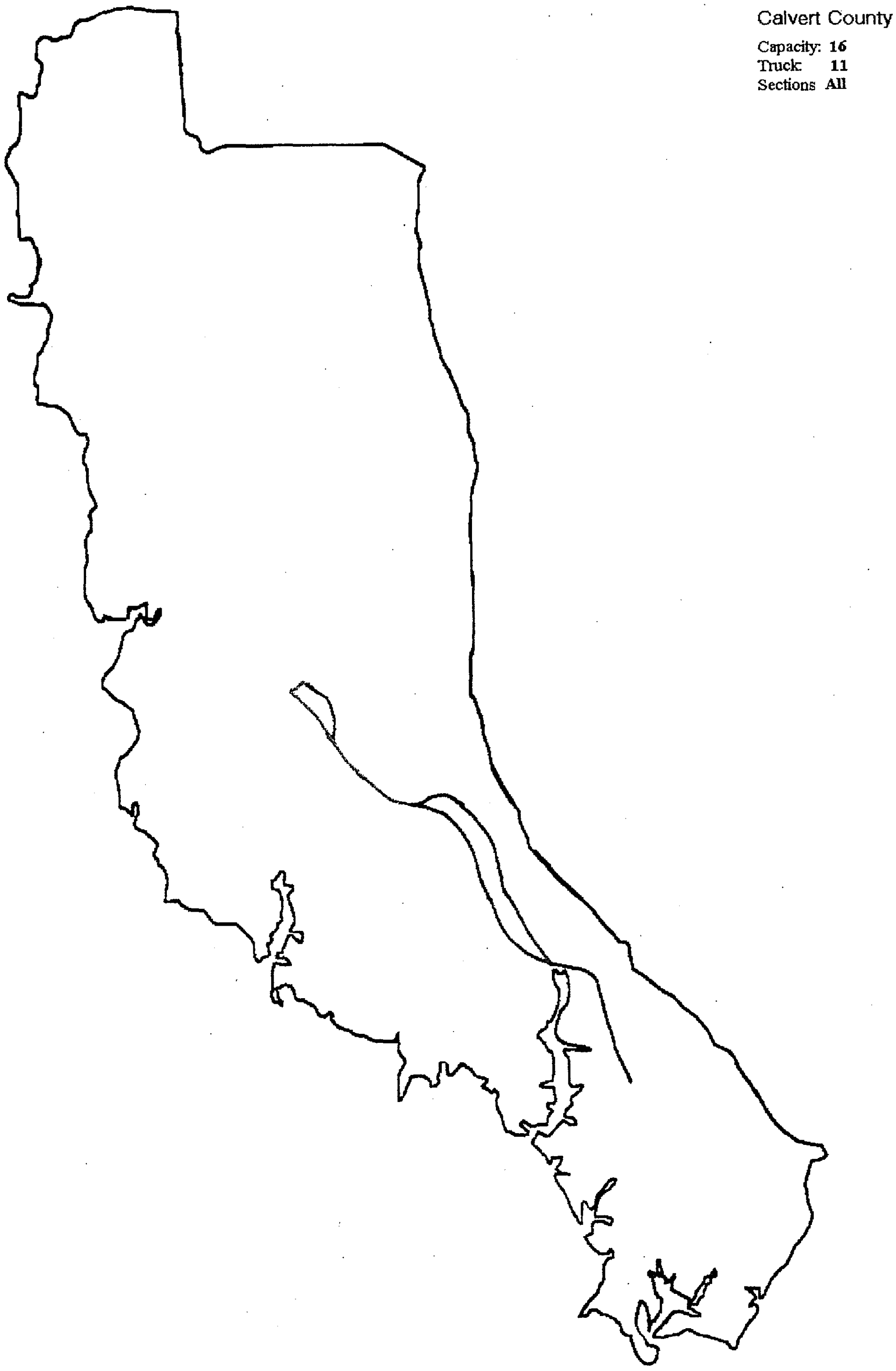


Figure 4.37 All Sections, Service Route Length 16 miles

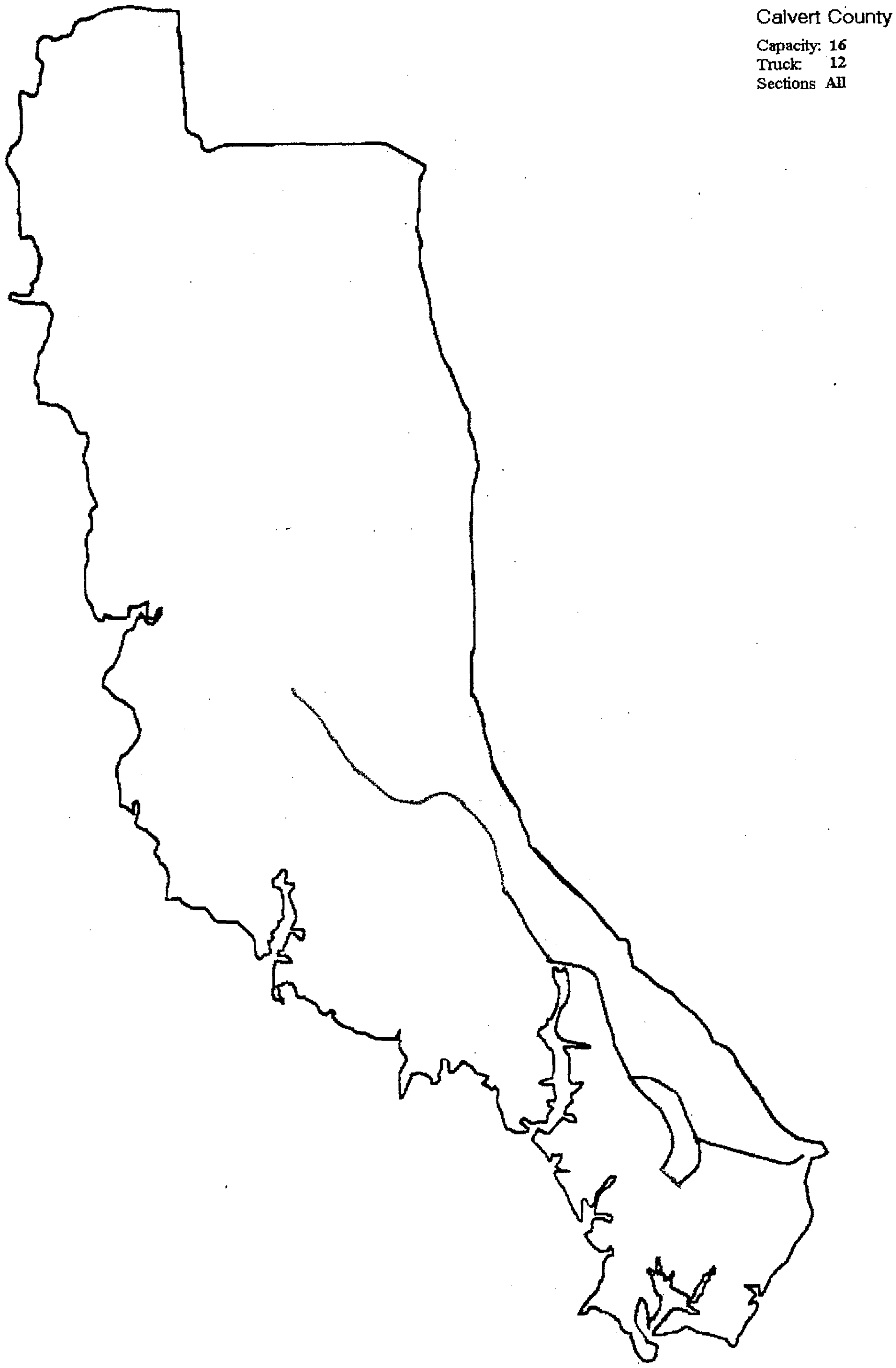


Figure 4.38 All Sections, Service Route Length 16 miles

Table 4.2

Summary of Result for Sections 1&2

Service Route Length 16 Miles

	Total	Service	Deadhead
Route 1:	31.28	12.17	19.11
Route 2:	22.90	10.78	12.12
Route 3:	49.87	15.98	33.89
Route 4:	42.85	14.94	27.91
Route 5:	40.84	15.16	25.68
Total	187.74	69.03	118.71
Average	37.55	13.81	23.74

Service Route Length 18 Miles

	Total	Service	Deadhead
Route 1:	45.59	17.56	28.03
Route 2:	29.72	17.26	12.46
Route 3:	21.80	16.50	5.30
Route 4:	29.83	17.71	12.12
Total	126.94	69.03	57.91
Average	31.74	17.26	14.48

Service Route Length 20 Miles

	Total	Service	Deadhead
Route 1:	32.02	18.55	13.47
Route 2:	33.90	19.44	14.46
Route 3:	39.04	17.84	21.20
Route 4:	21.80	13.20	8.60
Total	126.76	69.03	57.73
Average	31.69	17.26	14.43

Table 4.3

Summary of Result for Sections 3&4

Service Route Length 16 Miles

	Total	Service	Deadhead
Route 1:	51.76	15.76	36.00
Route 2:	35.02	13.16	21.86
Route 3:	36.36	15.76	20.60
Route 4:	17.47	13.29	4.18
Route 5:	26.74	15.43	11.31
Route 6:	28.01	15.25	12.76
Route 7:	6.94	6.94	0.00
Total	202.30	95.59	106.71
Average	28.90	13.66	15.24

Service Route Length 18 Miles

	Total	Service	Deadhead
Route 1:	16.34	11.64	4.70
Route 2:	36.33	17.05	19.28
Route 3:	51.76	17.97	33.79
Route 4:	16.33	13.93	2.40
Route 5:	24.22	17.11	7.11
Route 6:	44.93	17.89	27.04
Total	189.91	95.59	94.32
Average	31.65	15.93	15.72

Service Route Length 20 Miles

	Total	Service	Deadhead
Route 1:	54.02	19.11	34.91
Route 2:	35.71	19.97	15.74
Route 3:	44.16	18.15	26.01
Route 4:	25.71	18.97	6.74
Route 5:	31.80	19.39	12.41
Total	191.40	95.59	95.81
Average	38.28	19.12	19.16

Table 4.4

Summary of Result for All Sections

Service Route Length 16 Miles

	Total	Service	Deadhead
Route 1:	40.42	15.62	24.80
Route 2:	22.89	14.05	8.84
Route 3:	40.71	15.45	25.26
Route 4:	9.93	5.03	4.90
Route 5:	31.15	14.94	16.21
Route 6:	49.62	15.78	33.84
Route 7:	16.33	11.22	5.11
Route 8:	52.16	15.96	36.20
Route 9:	35.02	13.16	21.86
Route 10:	17.84	13.69	4.15
Route 11:	25.61	14.42	11.19
Route 12:	37.86	15.30	22.56
Total	379.54	164.62	214.92
Average	31.63	13.72	17.91

Service Route Length 18 Miles

	Total	Service	Deadhead
Route 1:	44.16	17.12	27.04
Route 2:	23.48	13.94	9.54
Route 3:	16.33	13.93	2.40
Route 4:	24.22	17.11	7.11
Route 5:	44.62	17.68	26.94
Route 6:	54.02	17.51	36.51
Route 7:	34.10	16.30	17.80
Route 8:	36.66	17.58	19.08
Route 9:	32.03	17.29	14.74
Route 10:	44.37	16.16	28.21
Total	353.99	164.62	189.37
Average	35.40	16.46	18.94

Table 4.4 (Cont'd)

Summary of Result for All Sections

Service Route Length 20 Miles

	Total	Service	Deadhead
Route 1:	26.26	16.25	10.01
Route 2:	54.02	19.11	34.91
Route 3:	38.96	19.25	19.71
Route 4:	37.56	17.68	19.88
Route 5:	34.94	17.99	16.95
Route 6:	40.71	19.43	21.28
Route 7:	36.77	19.48	17.29
Route 8:	25.71	19.56	6.15
Route 9:	35.02	15.87	19.15
Total	329.95	164.62	165.33
Average	36.66	18.29	18.37

Table 4.5

Comparison of the Results from the System Implementation and Current Service for Sections 1 & 2

	No. of Trucks	Total Route Length	Max Route Length	Min Route Length	Average Route Length	Total Deadhead	Average Deadhead
Current	6	172.63	36.18	22.42	28.77	104.94	17.49
16	5	187.74 8.75%	49.87 37.84%	22.90 2.14%	37.55 30.51%	118.71 13.12%	23.74 35.75%
18	4	154.70 -10.39%	52.53 45.19%	28.74 28.19%	38.68 34.43%	85.67 -18.36%	21.42 22.46%
20	4	154.52 -10.49%	45.98 27.09%	28.74 28.19%	38.63 34.27%	85.49 -18.53%	21.37 22.20%

Table 4.6

Comparison of the Result from the System Implementation and Current service for Sections 3 & 4

	No. of Trucks	Total Route Length	Max Route Length	Min Route Length	Average Route Length	Total Deadhead	Average Deadhead
Current	8	228.07	53.12	15.46	28.51	138.77	17.35
16	7	202.30 -11.30%	51.76 -2.56%	6.94 -55.11%	28.90 1.37%	106.71 -23.10%	15.24 -12.14
18	6	189.91 -16.73%	51.76 -2.56%	16.33 5.63%	31.65 11.02%	94.32 -32.03%	15.72 -9.39%
20	5	191.40 -16.08%	54.02 1.69%	25.71 66.30%	38.28 34.27%	95.81 -30.96%	19.16 10.44%

Table 4.7

Comparison of the Result from the System Implementation and Current service for Sections 1, 2, 3 & 4

	No. of Trucks	Total Route Length	Max Route Length	Min Route Length	Average Route Length	Total Deadhead	Average Deadhead
Current	14	400.70	53.12	15.46	28.62	243.71	17.41
16	12	379.54 -5.28%	52.16 -1.81%	9.93 -35.77%	31.63 10.51%	214.92 -11.81%	17.91 2.87%
18	10	353.99 -11.66%	54.02 1.69%	16.33 5.63%	35.40 23.69%	189.37 -22.30%	18.94 8.77%
20	9	329.95 -17.66%	54.02 1.69%	25.71 66.30%	36.66 28.10%	165.33 -32.16%	18.37 5.51%

REFERENCES

ALFA, A. S., and D. Q. Liu, 1988, "Postman Routing Problem in a Hierarchical Network," Engineering Optimization, 14, pp. 127-138.

Belenguer, H. M. and E. Benavent, 1991, "Polyhedral Results on the Capacitated Arc Routing Problem," Working paper Departamento De Estatdistica e Investigacion Opreativa, Universidad de VaLencia, Spain.

Campbell, J.F., and Andre Langevin, 1995, "Operations Management for Urban Snow Removal and Disposal". Transportation Research A. pp. 359-369.

Chapleau, L., J, A. Ferland, G. Lapaleme and J. Rousseau, 1984, "A Parallel Insert Method for the Capacitated Arc Routing Problem," Operations Research Letters, 3, pp. 95-99.

Christorfides, N, 1973, "The Optimum Traversal of a Graph," Omega 1, pp. 719-732.

Clark, G. and J.W. Wright, 1964, "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points," Operations Research, 12, pp. 568-581.

Cook, T.M., and B.S. Alprin, 1976, "Snow and Ice Removal in Urban Environment," Management Science, pp. 227-234.

Dror, M., H. Stern and Trudeau, 1987, "Postman Tour on a graph with precedence Relation on Arcs," Networks, pp. 283-294.

Desrochers, M., J. Desrosiers, and M.M. Solomon, 1992, "A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows," Operations Research, 40, pp. 342 - 354.

Eiselt, H.A., M. Gendreau, and G. Laporte, 1994, "Arc Routing Problems, Part I: the Rural Postman Problem", Operations Research, pp. 229-242.

Eiselt, H.A., M. Gendreau, and G. Laporte, 1994, "Arc Routing Problems, Part II: The Rural Postman Problem," Operation Research, pp. 399-413.

Eglese, R.W., 1994, "Routing Winter Gritting Vehicles," Discrete Applied Mathematics, pp. 231-244.

England, R., 1982, "Computer Analysis Ensures a Clean Sweep," Surveyor 6, May, 15.

Federal Highway Administration, 1997, Summary of SHRP Research and Economic Benefits of Snow and Ice Control.

Fisher, Marshall, 1994, "Solution of Vehicle Routing Problems Using Minimum K-Trees," Operations Research, 42, pp. 626-642.

Floyd, R.W. 1962, "Algorithm 97," Shortest Path Commun, Acm5, 345.

Gelinas, E, 1992, "Le Probleme du Postier Chinois Avec Constraites Generales de Preseance," M. Sc. A Dissertation, Ecole Polytechnique de Montreal.

Gilbert, J., 1990, "Fenerateur D'itineraires D'enlevement de la Neige", Publication CRT-648, Centre for Research on Transportation, University of Montreal.

Glover, F., 1997, "Tabu Search," Kluwer Academic Publishers, Boston.

Glover, F., and M. Laguna, 1993, "Tabu Search," in Modern Heuriestic Techniques for Combinatorial Problems, edited by C. Reeves, Blackwell, Oxford, pp77-150.

Goldberg, D.E., and Lingle R., 1985, "Loci and the Travelling Salesman Problem," Proceedings of an International Conference on Genetic Algorithms and Their Applications, Lawrence Erlbaum Associates, Hillsdale, NJ.

Golden, B. L., J. S. Dearmon and E. K. Baker, 1983,"Computational Experiments with Algorithms for a Class of Routing Problem," Computers Operations Research, pp. 47-59.

Golden, B. L. and R. T. Wong, 1981, "Capacitated Arc Routing Problems," Networks, pp. 305-315.

Guan, M., 1962, "Graphic Programming Using Odd and Even Points," Chinese Mathematics, 1, pp. 273- 277.

Haslam, E., and J.R. Wright, 1991, "Application of Routing Technologies to Rural Snow and Ice Control," Transortion Research Record , 1304, pp. 202-211.

Holland, J.H., 1975, "Adaptation in Natural and Artificial Systems," University of Michigan Press, Ann Arbor.

Kirkpatrick S., C. D. Gellat, and M. P. Vecchi, 1983, "Optimization by Simulated Annealing," Seience, 220, pp. 671-680.

Lenhart, M. M., 1998, "Vehicle Routing Problem for Snow and Ice Control Applications," M.S. Scholarly Paper, University of Maryland, College Park, MD.

Li, Chung-Lun, D. Simchi-Levi and M. Desrochers, 1992, "On the Distance Constrained Vehicle Routing Problem," Operations Research, pp. 790-799.

Liebling, T.M., 1973, "Routing Problems for Street Cleaning and Snow Removal," In R. Deininger (ed.), Models for Environmental Pollution Control, pp. 363-374, Ann Arbor Publishers.

Marks, H.D., and R. Stricker, 1971, "Routing for Public Service Vehicles," ASCE Jurnal of the Urban Planning and Development Division 97, pp. 16-178.

Metropolis, N., A. W. Rosenbluth, A. H. Teller, and E. Teller, 1953, "Equation of State Calculation by Fast Computing Machines", H. Of Chem . Phys., 21, pp. 1087-1091.

Or, I., 1976, "Traveling Salesman - type Combinatorial Problems and their relation to the Logistics of Blood Banking," Ph.D. Dissertation, Department of Industrial Engineering and Management Science, Northwestern University Evanston, IL.

Pearn, W. L., 1984, "The Capacitated Chinese Postman Problem," Ph.D. Dissertation, University of Maryland, College Park, MD.

Pearn, W. L., 1989, "Approximate Solutions for the Capacitated Arc Routing Problem," Computers Operations Research, pp. 589–600.

Pearn, W. L., 1991, "Augment-Insert Algorithms for the Capacitated Arc Routing Problem," Computers Operations Research, pp. 189–198.

Potvin, J.Y., and J. M. Rousseau, 19XX, "A Parallel Route building Algorithm for the Vehicle Routing and Scheduling Problem with Time Windows", European Journal of Operational Research, 66, pp. 331-340.

Rayward-Smith, V. J., I.H. Osman, C.R. Reeves and G.D. Smith, 1996, "Modern Heuristic Search Method,"

Solomon, M.M., 1987, "Algorithms for Vehicle Routing and Scheduling Problems with Time Window Constraints," Operations Research, 35, pp. 254-265.

Waddell, B., 1994, "Snow and Ice Control Excellence with Routing Software," Public Works, pp. 72 – 74.

Wang, Jin-Yuan, P. Kandula, and J. R. Wright, 1995, "Evaluation of Computer-Generated Routes for Improved Snow and Ice Control," Transportation Research Record 1509, pp 15 – 21.

Win, Z., 1987, "Contributions to Routing Problems," Ph.D. Dissertation, University of Augsburg, Germany.